L 12

```
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS   YYY         YYY     SSSSSSSSSSSS
SSS               YYY         YYY  SSS
SSS               YYY         YYY  SSS
SSS               YYY         YYY  SSS
SSS                  YYY   YYY     SSS
SSS                  YYY   YYY     SSS
SSS                  YYY   YYY     SSS
   SSSSSSSSS            YYY           SSSSSSSSS
   SSSSSSSSS            YYY           SSSSSSSSS
   SSSSSSSSS            YYY           SSSSSSSSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
            SSS         YYY                    SSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
SSSSSSSSSSSS            YYY        SSSSSSSSSSSS
```

```
_$

Ps
--

YZ

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$
```

```
**FILE**ID**SYSGETSYI                                   G 7
```

```
  SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  YY      YY    IIIIII
  SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  YY      YY    IIIIII
SS          YY      YY  SS          GG          EE              TT      SS          YY      YY      II
SS          YY      YY  SS          GG          EE              TT      SS          YY      YY      II
SS              YY  YY      SS          GG          EE              TT      SS              YY  YY          II
SS              YY  YY      SS          GG          EE              TT      SS              YY  YY          II
  SSSSSS        YY          SSSSSS    GG          EEEEEEEE        TT        SSSSSS        YY          II
  SSSSSS        YY          SSSSSS    GG          EEEEEEEE        TT        SSSSSS        YY          II
        SS      YY              SS  GG  GGGGGG  EE              TT              SS      YY          II
        SS      YY              SS  GG  GGGGGG  EE              TT              SS      YY          II
        SS      YY              SS  GG      GG  EE              TT              SS      YY          II
        SS      YY              SS  GG      GG  EE              TT              SS      YY          II          ....
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      SSSSSSSS        YY        IIIIII        ....
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      SSSSSSSS        YY        IIIIII        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II        SS
LL              II        SS
LL              II        SS
LL              II        SS
LL              II          SSSSSS
LL              II          SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

SYS
V04

```
0000        .TITLE  SYSGETSYI - GET SYSTEM INFORMATION SYSTEM SERVICE
0000        .IDENT  'V04-000'
0000   ;
0000   ;****************************************************************************
0000   ;*                                                                          *
0000   ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000   ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000   ;*  ALL RIGHTS RESERVED.                                                    *
0000   ;*                                                                          *
0000   ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000   ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000   ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000   ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000   ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000   ;*  TRANSFERRED.                                                            *
0000   ;*                                                                          *
0000   ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000   ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000   ;*  CORPORATION.                                                            *
0000   ;*                                                                          *
0000   ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000   ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000   ;*                                                                          *
0000   ;*                                                                          *
0000   ;****************************************************************************
0000
0000   ;++
0000   ; FACILITY: VMS Executive, System services.
0000   ;
0000   ; ABSTRACT:
0000   ;
0000   ;       Return processor information to caller, specifically
0000   ;       processor ID register, processor type, and VMS version number.
0000   ;
0000   ; ENVIRONMENT: Kernel Mode
0000   ;
0000   ; AUTHOR: John A. Ywoskus,       CREATION DATE: 06-August-1981
0000   ;
0000   ; MODIFIED BY:
0000   ;
0000   ;       V03-026 CWH3026         CW Hobbs                23-Jul-1984
0000   ;               Treat the QUANTUM item as special, since it is
0000   ;               stored as a negative number.
0000   ;
0000   ;       V03-025 MSH0059         Michael S. Harvey       3-Jul-1984
0000   ;               Treat a specified CSID argument of zero as if the argument
0000   ;               hadn't been specified at all. In either case, there is no
0000   ;               CSID value specified and the behavior of GETSYI should be
0000   ;               the same for both.
0000   ;
0000   ;       V03-024 MSH0021         Michael S. Harvey       9-Mar-1984
0000   ;               Allow access to SYI items stashed away in the local SB,
0000   ;               regardless of whether we're in a cluster or not.
0000   ;
0000   ;       V03-023 MSH0013         Michael S. Harvey       2-Mar-1984
0000   ;               Correctly extract node name length so as not to clobber
0000   ;               P1 space and crash the system.
```

```
0000  ;
0000  ;       V03-022 WMC0003         Wayne Cardoza           9-Feb-1984
0000  ;               Add $ARCDEF.
0000  ;
0000  ;       V03-021 WMC0002         Wayne Cardoza           29-Jan-1984
0000  ;               Add F and G floating flags.
0000  ;
0000  ;       V03-020 KPL0001         Peter Lieberwirth       15-Jan-1984
0000  ;               Fix typo in V03-019.
0000  ;
0000  ;       V03-019 WMC0001         Wayne Cardoza           07-Jan-1984
0000  ;               Add page and swap file sizes.
0000  ;
0000  ;       V03-018 TCM0001         Trudy C. Matthews       28-Dec-1983
0000  ;               In EXE$NAMCSID, do not access node name passed by caller
0000  ;               after raising IPL (it may be pagable).  Make QUORUM a
0000  ;               special parameter; it is stored as a negative value but should
0000  ;               be displayed as a positive one.
0000  ;
0000  ;       V03-017 KFH0011         Ken Henderson           30 Aug 1983
0000  ;               Fix resetting of IPL on error path.
0000  ;               Add documentation of how
0000  ;               itemcodes are added.
0000  ;
0000  ;       V03-016 KFH0010         Ken Henderson           23 Aug 1983
0000  ;               Fix checking of item code validity.
0000  ;               Update max structure code.
0000  ;
0000  ;       V03-015 KFH0009         Ken Henderson           18 Aug 1983
0000  ;               Made SCS_EXISTS special and boolean.
0000  ;
0000  ;       V03-014 KFH0008         Ken Henderson           28 Jul 1983
0000  ;               Finished support for 'retired' item-codes.
0000  ;               Took out call to SCS$CONFIG_SYS.
0000  ;
0000  ;       V03-013 KFH0007         Ken Henderson           12 Jul 1983
0000  ;               Added temporary additional check for
0000  ;               clusterness.
0000  ;
0000  ;       V03-012 KFH0006         Ken Henderson           26 May 1983
0000  ;               Changed EXE$NAMCSID entry point to
0000  ;               be non-Global.
0000  ;
0000  ;       V03-011 KFH0005         Ken Henderson           25 May 1983
0000  ;               Updated code to use IFCLSTR and IFNOCLSTR.
0000  ;
0000  ;       V03-010 KFH0004         Ken Henderson           21 May 1983
0000  ;               Added support for wild-carding through
0000  ;               nodes. Added NAMCSID and EXE$NAMCSID routines.
0000  ;               Cleaned up usage of LOCAL_SPACE on stack.
0000  ;
0000  ;       V03-009 KFH0003         Ken Henderson           11 Mar 1983
0000  ;               Added .WARN if item-code is undefined.
0000  ;
0000  ;       V03-008 KFH0002         Ken Henderson           25 Feb 1983
0000  ;               Added definition of GETSYISW.
0000  ;
```

```
0000  ;       V03-007 KFH0001         Ken Henderson           16 Feb 1983
0000  ;               Major rewrite of EXE$GETSYI and related
0000  ;               routines to make it table-driven like GETJPI,
0000  ;               and allow for SYSBOOT parameters and other
0000  ;               enhancements.
0000  ;
0000  ;       V03-006 MSH0001         Maryann Hinden          23-Mar-1982
0000  ;               Fix broken BSBW.
0000  ;
0000  ;       V03-005 JAY0006         John A. Ywoskus         17-Mar-1982
0000  ;               Change SS$_EXQUOTA return error to SS$_EXASTLM.
0000  ;
0000  ;       V03-004 JAY0005         John A. Ywoskus         21-Jan-1982
0000  ;               Return 8 bytes for system version instead of 4.
0000  ;               General cleanup.
0000  ;
0000  ;       V02-003 LJK0082         Lawrence J. Kenah       11-Nov-1981
0000  ;               Write accessibility of multiple page buffer can
0000  ;               now be done on global routine.
0000  ;
0000  ;       V03-002 JAY0004         John A. Ywoskus         05-Oct-1981
0000  ;               Add null arguments so call list is compatable with
0000  ;               $GETJPI. Also, make external references be
0000  ;               addressed with G^, and include VA and PSL defs.
0000  ;
0000  ;       V03-001 JAY0003         John A. Ywoskus         08-Sep-1981
0000  ;               Fix null item bug, make return length optional.
0000  ;--
0000  ;
```

```
0000  ;                          GUIDE TO GETJPI/GETSYI/GETDVI
0000  ;                          -----------------------------
0000  ;
0000  ;Overview
0000  ;--------
0000  ;
0000  ;These three system services are table-driven. The macro definition files
0000  ;that help define their tables are shared with DCL and the RTL. This results
0000  ;in new item-codes becoming useable with DCL's F$GETXXI lexical functions and
0000  ;the RTL's LIB$GETXXI routines automatically. Additionally, new SYSBOOT
0000  ;parameters become item-codes to the GETSYIs.
0000  ;
0000  ;The macro definition files are called JPITABLE.MAR, SYITABLE.MAR, and
0000  ;DVITABLE.MAR, and live in MASD$:<VMSLIB.SRC>. During a systembuild, they
0000  ;are inserted into the library SYS$LIBRARY:SYSBLDMLB.MLB. DCL and the RTL
0000  ;and SYS use this library to define their GETXXI tables. The system
0000  ;parameter file <SYS.SRC>SYSPARAM.MAR has also been conditionalized to be
0000  ;used to define GETSYI item-codes and is also inserted into SYSBLDMLB.MLB.
0000  ;
0000  ;
0000  ;NOTE:   SYSBLDMLB.MLB is a general macro library for holding macro
0000  ;        definitions that are shared between facilities, but will not
0000  ;        ship to the customer.
0000  ;
0000  ;
0000  ;When adding an item-code, at least two files need to be edited. One of the
0000  ;macro files listed above, as well as an SDL file that defines the 16-bit
0000  ;number which is the user-visible item-code. Also, if a SYSBOOT parameter is
0000  ;added, an SDL file needs to be updated to define the new GETSYI item-code.
0000  ;
0000  ;The GETDVI service actually uses only one table, but the GETSYI and GETJPI
0000  ;services use several. The JPITABLE file defines all the tables for GETJPI
0000  ;and the SYITABLE file defines all the tables for GETSYI. The different
0000  ;tables group the pieces of data according to method of retrieval.
0000  ;
0000  ;In some cases, the piece of data to be returned by the service requires
0000  ;special processing to fetch, calculate, or format it before returning it.
0000  ;In these cases, the code of the system service needs to be enhanced.
0000  ;If the data returned is a new format for DCL, the lexical function
0000  ;module of DCL may need to be enhanced. This is also true for the RTL code.
```

```
0000  ;The Macros
0000  ;----------
0000  ;
0000  ;A two-level scheme exists for defining the item tables used by the three
0000  ;services and the other facilities. A commonly defined macro (called
0000  ;JPI_GENERATE_TABLE, SYI_GENERATE_TABLE, or DVI_GENERATE_TABLE) contains
0000  ;multiple calls to a lower-level macro (called JPI_ITEM_CODE, SYI_ITEM_CODE,
0000  ;or DVI_ITEM_CODE) which actually defines each element in the table.
0000  ;While the _GENERATE_TABLE macros are commonly defined, the _ITEM_CODE macros
0000  ;are individually defined according to the needs of facility. (For instance,
0000  ;the LEXICON module must store the name of the item as an ASCIC string - in
0000  ;order to match it with the string supplied in the F$GETXXI function call;
0000  ;the other facilities need not store the item name in text.)
0000  ;
0000  ;When an item-code must be added, an additional call to the _ITEM_CODE macro
0000  ;must be added to the appropriate _GENERATE_TABLE macro. In the case of GETJPI
0000  ;and GETDVI, the _GENERATE_TABLE macro is defined in the JPITABLE and DVITABLE
0000  ;modules. The SYI_GENERATE_TABLE macro is defined by the SYSPARAM module
0000  ;- all the calls to the PARAMETER and PQL macros are 'collected' into the
0000  ;SYI_GENERATE_TABLE macro. When used in that mode (when GETSYISW is defined),
0000  ;the SYI_ITEMTABLES macro also becomes part of the SYI_GENERATE_TABLE macro.
0000  ;SYI_ITEMTABLES is defined in the SYITABLE module and contains all the calls
0000  ;to the SYI_ITEM_CODE macro that are Not related to SYSBOOT parameters.
0000  ;When GETSYISW is defined in SYSPARAM, the PARAMETER macro does not allocate
0000  ;or store memory, but rather passes some of the arguments to it on through via
0000  ;a call to SYI_ITEM_CODE. That is how all the calls to PARAMETER become calls
0000  ;to SYI_ITEM_CODE.
0000  ;
0000  ;The following is the situation that exists when the symbol GETSYISW is defined.
0000  ;The non-SYSBOOT items are defined by the macro SYI_ITEMTABLES in SYITABLE.MAR.
0000  ;The SYSBOOT items are defined by each invokation of the PARAMETER macro in
0000  ;SYSPARAM.MAR. Note that each invokation of the PQL macro in SYSPARAM.MAR
0000  ;invokes the PARAMETER macro twice. When GETSYISW is defined, the PARAMETER
0000  ;macro merely passes its arguments through to a call to the SYI_ITEM_CODE
0000  ;macro. The SYI_ITEM_CODE macro is locally defined as needed by the facility.
0000  ;
0000  ;+---------------------------------------------------------------------+
0000  ;|                    SYI_GENERATE_TABLE                               |
0000  ;|   SYI_ITEMTABLES          |                                         |
0000  ;|                           | PARAMETER   |           PQL             |
0000  ;|                           |             | PARAMETER   | PARAMETER   |
0000  ;|             |             |             |             |             |
0000  ;|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|
0000  ;+---------------------------------------------------------------------+
0000  ;\                          / \                                        /
0000  ; \                        /   \                                      /
0000  ;     FROM SYITABLE.MAR                  FROM SYSPARAM.MAR
0000  ;    (NON-SYSBOOT ITEMS)                  (SYSBOOT ITEMS)
```

```
                  0000            .SBTTL  DECLARATIONS
                  0000            $ARCDEF                               ; architectural flags
                  0000            $CLUBDEF                              ; cluster block definitions
                  0000            $CSBDEF                               ; cluster system block definitions
                  0000            $IPLDEF                               ; IPL definitions
                  0000            $PCBDEF                               ; define processor control block
                  0000            $PFLDEF                               ; page file control block
                  0000            $PRDEF                                ; define processor registers
                  0000            $PSLDEF                               ; define processor status register
                  0000            $SBDEF                                ; system block definitions
                  0000            $SSDEF                                ; define status codes
                  0000            $SYIDEF                               ; define GETSYI item identifiers
                  0000            $VADEF                                ; virtual addressing definitions
                  0000
                  0000    ;
                  0000    ; Define the following symbol so that SYSPARAM macros will conditionalize
                  0000    ; correctly for us.
                  0000    ;
        00000000  0000            GETSYISW = 0
                  0000
                  0000    ;
                  0000    ; MACROS:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; Macros to define entries in the four item information tables.
                  0000    ; There is a table for each data structure from which the user may
                  0000    ; request information, and one table for information returned as an
                  0000    ; address.  Tables are indexed by low byte of item identifier.
                  0000    ; Refer to "OWN STORAGE:" for pictures of the table entries.
                  0000    ;
                  0000
                  0000            .MACRO  SYI_ITEM_CODE   BASE,-          ; for service to use
                  0000                                    NAME,-          ; of the item-code
                  0000                                    SOURCE,-        ; of the data
                  0000                                    DTYPE,-         ; of returned value
                  0000                                    BITPOS,-        ; of FLD type data
                  0000                                    BITSIZ,-        ; "
                  0000                                    OUTLEN          ; of returned value
                  0000
                  0000            .IF NOT_DEFINED SYI$_'NAME
                  0000
                  0000            .IF IDENTICAL <BASE><EXE>
                  0000
                  0000            .WARN ; SYI$_'NAME IS NOT DEFINED AS 'EXE' IN STARDEFQZ.SDL
                  0000
                  0000            .ENDC   ; IDENTICAL
                  0000
                  0000            .IF IDENTICAL <BASE><FLD>
                  0000
                  0000            .WARN ; SYI$_'NAME IS NOT DEFINED AS 'FLD' IN STARDEFQZ.SDL
                  0000
                  0000            .ENDC   ; IDENTICAL
                  0000
                  0000            .ENDC   ; NOT_DEFINED
                  0000
                  0000            STEP = 5
```

```
          0000        .IIF IDENTICAL <BASE><EXE>,      STEP = 5
          0000        .IIF IDENTICAL <BASE><FLD>,      STEP = 7
          0000
          0000        XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><HEXNUM>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><DECNUM>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><PRVMSK>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><PADSTR>, XTYPE = BSTRING
          0000        .IIF IDENTICAL <DTYPE><HEXSTR>, XTYPE = BSTRING
          0000        .IIF IDENTICAL <DTYPE><CNTSTR>, XTYPE = CSTRING
          0000        .IIF IDENTICAL <DTYPE><STRDSC>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><BITVEC>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><BITVAL>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><STDUIC>, XTYPE = VALUE
          0000        .IIF IDENTICAL <DTYPE><STDTIM>, XTYPE = VALUE
          0000
          0000        . = BASE'TBL + <<SYI$_'NAME & ^XFFF> * STEP>
          0000
          0000        .IIF    IDENTICAL <BASE><FLD>,  .WORD   <BITSIZ-1>@11!BITPOS
          0000
          0000        .LONG   SOURCE
          0000        .BYTE   XTYPE@5!OUTLEN
          0000
          0000        .ENDM   SYI_ITEM_CODE
          0000
          0000  ;
          0000  ; This macro defines the entries to the table of special items.
          0000  ; The items in this table must be handled by action routines
          0000  ; before being returned.  Each entry has a word item identifier
          0000  ; followed by the address of an action routine.
          0000  ; ALL PROCESSOR REGISTER ITEMS ARE SPECIALS.
          0000  ;
          0000
          0000        .MACRO  SPECIAL_ITEM NAME,ROUTINE
          0000        .WORD   SYI$_'NAME
          0000        .ADDRESS ROUTINE
          0000        .ENDM   SPECIAL_ITEM
          0000
          0000  ;
          0000  ; This macro defines flag bits.
          0000  ;
          0000
          0000        .MACRO  SYIBITS NAME,SIZE
          0000        SYI_V_'NAME' = SYI_BIT
          0000        SYI_S_'NAME' = SIZE
          0000        SYI_BIT = SYI_BIT + SIZE
          0000        .ENDM   SYIBITS
          0000
          0000  ;
          0000  ; EQUATED SYMBOLS:
          0000  ;
          0000
00000004  0000        EFN      = 4                     ; event flag number argument
00000008  0000        NULLARG1 = 8                     ; first null argument
0000000C  0000        NULLARG2 = 12                    ; second null argument
00000010  0000        ITMLST   = 16                    ; address of item identifiers
00000014  0000        IOSB     = 20                    ; I/O status block address
```

```
00000018  0000          ASTADR   = 24                       ; ast routine address
0000001C  0000          ASTPRM   = 28                       ; ast parameter
00000002  0000          MAXSTRUC = 2                        ; maximum structure code
00000000  0000          VALUE    = 0                        ; datatypes
00000001  0000          BSTRING  = 1
00000002  0000          CSTRING  = 2
FFFFFFE0  0000          LOCAL_SPACE = -32                   ; 8 longwords on stack
FFFFFFE0  0000          BITSIZ   = LOCAL_SPACE+0
FFFFFFE4  0000          BITPOS   = LOCAL_SPACE+4
FFFFFFE8  0000          TEMPORARY = LOCAL_SPACE+8
FFFFFFEC  0000          SPECIAL_SPACE = LOCAL_SPACE+12
FFFFFFFC  0000          FLAGS    = LOCAL_SPACE+28
          0000
          0000    ;
          0000    ; Bit definitions for flags longword on stack
          0000    ;
          0000
00000000  0000          SYI_BIT = 0
          0000          SYIBITS WILD,1                      ; we're doing a wildcard operation
          0000          SYIBITS INCLUSTER,1                 ; we're in a live cluster
          0000          SYIBITS REMOTE_NODE,1               ; the target node isn't the local node
          0000          SYIBITS RETIRED,1                   ; the item-code isn't in use anymore
          0000
          0000    ;
          0000    ; Max structure number definitions
          0000    ;
          0000
00000101  0000          MAX_EXE_ITEM = <SYI$_LASTEXE&^XFFF>-1   ; maximum EXE item number
0000002A  0000          MAX_FLD_ITEM = <SYI$_LASTFLD&^XFFF>-1   ; maximum FLD item number
          0000
          0000    ;
          0000    ; OWN STORAGE:
          0000    ;
          0000
      00000000          .PSECT  YF$$SYSGETSYI
          0000
          0000    ;
          0000    ; This array contains the maximum item number for each of the two
          0000    ; item data structures, indexed by structure number.
          0000    ;
          0000    MAXCOUNT:
    0101  0000          .WORD   MAX_EXE_ITEM
    002A  0002          .WORD   MAX_FLD_ITEM
          0004
```

```
        0004    ;
        0004    ; The following tables are zeroed explicitly to allow the code to
        0004    ; recognize an uninitialized element (because of a retired item-code)
        0004    ;
        0004
        0004    EXETBL:
        0004                ;-------------------------;
        0004                ; .LONG SOURCE            ;
        0004                ; .BYTE DTYPE@5!OUTLEN    ;
        0004                ;-------------------------;
        0004
        0004                .REPEAT 5*<MAX_EXE_ITEM+1>
        0004                .BYTE   0
    00  0004                .ENDR
        050E
        050E    FLDTBL:
        050E                ;------------------------------;
        050E                ; .WORD <BITSIZ-1>@11!BITPOS   ;
        050E                ; .LONG SOURCE                 ;
        050E                ; .BYTE DTYPE@5!OUTLEN         ;
        050E                ;------------------------------;
        050E
        050E                .REPEAT 7*<MAX_FLD_ITEM+1>
        050E                .BYTE   0
    00  050E                .ENDR
        063B
        063B                .SAVE
        063B
        063B    ;*************************************************************
        063B    ;
        063B    ; GENERATE THE TABLES USING THE COMMONLY DEFINED MACRO
        063B    ;
        063B    ;*************************************************************
        063B
        063B                SYI_GENERATE_TABLE
```

```
                  0243
                  0243          .NLIST  CND
                  0243          PARAMETER       ADDRESS=EXE$GL_DEFFLAGS,-
                  0243                          DEFAULT=1,-
                  0243                          MAX=1,-
                  0243                          MIN=0,-
                  0243                          NAME=BUGREBOOT,-
                  0243                          BIT=EXE$V_BUGREBOOT,-
                  0243                          TYPE=<DYNAMIC,SYS>,-
                  0243                          UNIT=Boolean
                  0243
        00000004  0243          OUTLEN = 4
                  0243          SYI_ITEM_CODE   FLD,-
                  0243                          <BUGREBOOT>,-
                  0243                          <EXE$GL_DEFFLAGS>,-
                  0243                          BITVAL,-
                  0243                          <EXE$V_BUGREBOOT>,-
                  0243                          1,-
                  0243                          1
                  0243
                  0243
        00000005  0243          STEP = 5
                  0243
        00000000  0243          XTYPE = VALUE
                  0243
        00000515  0243          . = FLDTBL + <<SYI$_BUGREBOOT & ^XFFF> * STEP>
                  0515
                  0517
       00000000'  0517          .LONG   EXE$GL_DEFFLAGS
              01  051B          .BYTE   XTYPE@5!1
                  051C
                  051C
                  051C
```

```
                     03F6
                 0000063B            .RESTORE
                     063B
                     063B   ;
                     063B   ; Table to define items which must be handled by action routines
                     063B   ;
                     063B
                     063B   SPECIAL:
                     063B            SPECIAL_ITEM    CLUSTER_MEMBER, SPC_MEMBER
                     0641            SPECIAL_ITEM    CLUSTER_NODES, SPC_CLUB
                     0647            SPECIAL_ITEM    CLUSTER_VOTES, SPC_CLUB
                     064D            SPECIAL_ITEM    CLUSTER_QUORUM, SPC_CLUB
                     0653            SPECIAL_ITEM    CLUSTER_FSYSID, SPC_CLUB
                     0659            SPECIAL_ITEM    CLUSTER_FTIME, SPC_CLUB
                     065F            SPECIAL_ITEM    NODE_CSID, SPC_CSB
                     0665            SPECIAL_ITEM    NODE_VOTES, SPC_CSB
                     066B            SPECIAL_ITEM    NODE_QUORUM, SPC_CSB
                     0671            SPECIAL_ITEM    NODE_SYSTEMID, SPC_SB
                     0677            SPECIAL_ITEM    NODE_AREA, SPC_SB
                     067D            SPECIAL_ITEM    NODE_NUMBER, SPC_SB
                     0683            SPECIAL_ITEM    NODE_SWINCARN, SPC_SB
                     0689            SPECIAL_ITEM    NODE_SWTYPE, SPC_SB
                     068F            SPECIAL_ITEM    NODE_SWVERS, SPC_SB
                     0695            SPECIAL_ITEM    NODE_HWTYPE, SPC_SB
                     069B            SPECIAL_ITEM    NODE_HWVERS, SPC_SB
                     06A1            SPECIAL_ITEM    NODENAME, SPC_SB
                     06A7            SPECIAL_ITEM    SCS_EXISTS, SPC_EXISTS
                     06AD            SPECIAL_ITEM    SID, SPC_PROCREG
                     06B3            SPECIAL_ITEM    CPU, SPC_PROCREG
                     06B9            SPECIAL_ITEM    PAGEFILE_PAGE, SPC_PAGESWAP
                     06BF            SPECIAL_ITEM    SWAPFILE_PAGE, SPC_PAGESWAP
                     06C5            SPECIAL_ITEM    PAGEFILE_FREE, SPC_PAGESWAP
                     06CB            SPECIAL_ITEM    SWAPFILE_FREE, SPC_PAGESWAP
                     06D1            SPECIAL_ITEM    QUANTUM, SPC_NEGATIVE
         0000001A    06D7   SPECIAL_LEN = <.-SPECIAL>/6
                     06D7
```

```
06D7          .SBTTL  SYSGETSYI - GETSYI main program
06D7
06D7    ;++
06D7    ;
06D7    ; FUNCTIONAL DESCRIPTION:
06D7    ;
06D7    ;       This service allows a process to receive status and identification
06D7    ;       information about the system on which the calling process is running.
06D7    ;
06D7    ; CALLING SEQUENCE:
06D7    ;
06D7    ;       CALLS/CALLG
06D7    ;
06D7    ; INPUTS:
06D7    ;
06D7    ;       EFN(AP) = number of the event flag to set when all of the requested
06D7    ;                 data is valid.
06D7    ;       NODE(AP) = pointer to nodename descriptor
06D7    ;       CSIDADR(AP) = address of CSID source/destination
06D7    ;       ITMLST(AP) = address of a list of item descriptors of the form:
06D7    ;
06D7    ;               +-------------------------------+
06D7    ;               !  ITEM CODE   !  BUF. LENGTH   !
06D7    ;               +-------------------------------+
06D7    ;               !        BUFFER ADDRESS         !
06D7    ;               +-------------------------------+
06D7    ;               !  ADDRESS TO RETURN LENGTH     !
06D7    ;               +-------------------------------+
06D7    ;
06D7    ;       IOSB(AP) = address of a quadword I/O status block to receive final
06D7    ;               status
06D7    ;       ASTADR(AP) = address of an AST routine to be called when all of the
06D7    ;               requested data has been supplied.
06D7    ;       ASTPRM(AP) = 32 bit ast parameter
06D7    ;
06D7    ; IMPLICIT INPUTS:
06D7    ;
06D7    ;       none
06D7    ;
06D7    ; OUTPUTS:
06D7    ;
06D7    ;       none
06D7    ;
06D7    ; IMPLICIT OUTPUTS:
06D7    ;
06D7    ;       none
06D7    ;
06D7    ; ROUTINE VALUE:
06D7    ;
06D7    ;       SS$_NORMAL -> normal completion
06D7    ;       SS$_EXASTLM -> AST quota exceeded
06D7    ;       SS$_ACCVIO -> ITMLST can not be read by the calling access mode,
06D7    ;                       or the return buffer or return length word can not
06D7    ;                       be written by the calling access mode
06D7    ;       SS$_BADPARAM -> an invalid item identifier was supplied
06D7    ;
06D7    ; SIDE EFFECTS:
```

```
                          06D7  ;
                          06D7  ;         none
                          06D7  ;--
                          06D7
                      00000000            .PSECT  YEXEPAGED           ; only entry mask in this program section
                          0000
                     0FFC 0000            .ENTRY  EXE$GETSYI,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
              06D2'    31 0002            BRW     EXE_GETSYI          ; transfer to real procedure
                          0005
                      000006D7            .PSECT  YF$$SYSGETSYI
                          06D7
                          06D7  ;
                          06D7  ; Allocate some local space on the stack
                          06D7  ;
                          06D7  EXE_GETSYI:
      5E   E0 AE       DE 06D7            MOVAL   LOCAL_SPACE(SP), SP
           FC AD       D4 06DB            CLRL    FLAGS(FP)                       ; reset the flags longword
            030D       30 06DE            BSBW    NAMCSID                         ; process nodename/CSID pair
           18 50       E9 06E1            BLBC    R0,4$
                          06E4
                          06E4  ;
                          06E4  ; Check for and clear possible IOSB
                          06E4  ;
      51   14 AC       D0 06E4            MOVL    IOSB(AP),R1                     ; get IOSB address
              08       13 06E8            BEQL    3$                              ; branch if none
                          06EA            IFNOWRT #8,(R1),30$                     ; check write access to it
              61       7C 06F0            CLRQ    (R1)                            ; clear IOSB
                          06F2  ;
                          06F2  ; Check for and clear event flag
                          06F2  ;
      53   04 AC       9A 06F2  3$:       MOVZBL  EFN(AP),R3                      ; get event flag number
   00000000'GF         16 06F6            JSB     G^SCH$CLREF                     ; clear event flag
           6B 50       E9 06FC  4$:       BLBC    R0,GRET                         ; and return on errors
                          06FF  ;
                          06FF  ; Validate AST, if present
                          06FF  ;
           18 AC       D5 06FF            TSTL    ASTADR(AP)
              0C       13 0702            BEQL    5$                              ; no AST to check
 54   00000000'GF      D0 0704            MOVL    G^CTL$GL_PCB,R4                 ; get our PCB address
           38 A4       B5 070B            TSTW    PCB$W_ASTCNT(R4)                ; is AST quota exceeded?
              4B       15 070E            BLEQ    40$                             ; branch if so and return error
                          0710  ;
                          0710  ; Loop through item descriptors, validating the requested item
                          0710  ; identifiers and moving accessible items. A zero item identifier
                          0710  ; terminates the list.
                          0710  ;
      55   10 AC       D0 0710  5$:       MOVL    ITMLST(AP),R5                   ; get item descriptor list address
              4C       13 0714            BEQL    50$                             ; ITMLST not optional
                          0716            IFNORD  #4,(R5),30$                     ; check first longword readable
                          071C  10$:                                              ; top of item-get loop
         56   85       3C 071C            MOVZWL  (R5)+,R6                        ; get buffer size
         51   85       3C 071F            MOVZWL  (R5)+,R1                        ; get item identifier
              43       13 0722            BEQL    60$                             ; done if zero, take normal exit
                          0724            IFNORD  #12,(R5),30$                    ; check rest of this descriptor ...
                          072A                                                    ; plus first longword of next one
         57   85       7D 072A            MOVQ    (R5)+,R7                        ; R7 = buffer address, R8 = length address
              51       DD 072D            PUSHL   R1                              ; save R1 across accessibility check
              50   57    D0  072F              MOVL     R7,R0                          ; buffer address to R0
              51   56    D0  0732              MOVL     R6,R1                          ; and size to R1
                   53    D4  0735              CLRL     R3                             ; PROBE will use PSL<PRVMOD>
         00000000'EF     16  0737              JSB      EXE$PROBEW                     ; check write accessibility of buffer
                16 50    E9  073D              BLBC     R0,30$                         ; buffer not accessible
                   51  8ED0  0740              POPL     R1                             ; restore R1 for use by CHECKITEM
                   55    DD  0743              PUSHL    R5                             ; save R5 across item check
                 0075    30  0745              BSBW     CHECKITEM                      ; check item's validity
                17 50    E9  0748              BLBC     R0,50$                         ; return error if not valid
                 00F8    30  074B              BSBW     PUTDATA                        ; put the item requested in user buffer
                   55  8ED0  074E              POPL     R5                             ; unsave R5
                C8 50    E8  0751              BLBS     R0,10$                         ; continue on success
                   14    11  0754              BRB      GRET
                             0756
                             0756      ;
                             0756      ; Error/success dispatch points:
                             0756      ;
              50   0C    3C  0756      30$:    MOVZWL   #SS$_ACCVIO,R0                 ; access violation
                   0F    11  0759              BRB      GRET                           ; terminate service below
        50    2A04 8F    3C  075B      40$:    MOVZWL   #SS$_EXASTLM,R0                ; AST quota exceeded
                   08    11  0760              BRB      GRET                           ; terminate service below
              50   14    3C  0762      50$:    MOVZWL   #SS$_BADPARAM,R0               ; illegal item or request
                   03    11  0765              BRB      GRET                           ; terminate service below
              50   01    3C  0767      60$:    MOVZWL   #SS$_NORMAL,R0                 ; normal return
                             076A      ;
                             076A      ; Set the event flag, post completion status, and declare completion AST
                             076A      ;
                   50    DD  076A      GRET:   PUSHL    R0                             ; save completion status
  54     00000000'GF     D0  076C              MOVL     G^CTL$GL_PCB,R4                ; get PCB address
           51   60 A4    D0  0773              MOVL     PCB$L_PID(R4),R1               ; get process's PID
                   52    D4  0777              CLRL     R2                             ; set null priority increment
           53   04 AC    9A  0779              MOVZBL   EFN(AP),R3                     ; get event flag number to set
         00000000'GF     16  077D              JSB      G^SCH$POSTEF                   ; set the event flag
           51   14 AC    D0  0783      10$:    MOVL     IOSB(AP),R1                    ; get address of IOSB
                   09    13  0787              BEQL     20$                            ; branch if none
                             0789              IFNOWRT  #8,(R1),20$                    ; check if writable
              61   6E    D0  078F              MOVL     (SP),(R1)                      ; store completion status
           55   18 AC    D0  0792      20$:    MOVL     ASTADR(AP),R5                  ; get address of AST routine
                   15    13  0796              BEQL     30$                            ; branch if none specified
                   54    DC  0798              MOVPSL   R4                             ; get PSL
  54   54   02     16    EF  079A              EXTZV    #PSL$V_PRVMOD,#PSL$S_PRVMOD,R4,R4 ; extract previous mode
                             079F              $DCLAST_S (R5),ASTPRM(AP),R4            ; queue the completion AST
                   50  8ED0  07AD      30$:    POPL     R0                             ; restore completion status
                         04  07B0              RET                                     ; and return.
                             07B1
```

```
07B1          .SBTTL  CHECKITEM - Validate item identifier
07B1
07B1    ;++
07B1    ;
07B1    ; FUNCTIONAL DESCRIPTION:
07B1    ;
07B1    ;       Routine to validate item identifier and return information
07B1    ;       about the item.
07B1    ;
07B1    ; CALLING SEQUENCE:
07B1    ;
07B1    ;       JSB/BSB
07B1    ;
07B1    ; INPUTS:
07B1    ;
07B1    ;       R1 = item identifier
07B1    ;
07B1    ; IMPLICIT INPUTS:
07B1    ;
07B1    ;       none
07B1    ;
07B1    ; OUTPUTS:
07B1    ;
07B1    ;       R1 = item identifier
07B1    ;       R2 = structure number
07B1    ;       R3 = item length
07B1    ;       R4 = item source address
07B1    ;       R5 = item type code
07B1    ;       BITSIZ(FP) - if FLD
07B1    ;       BITPOS(FP) - if FLD
07B1    ;
07B1    ; IMPLICIT OUTPUTS:
07B1    ;
07B1    ;       none
07B1    ;
07B1    ; ROUTINE VALUE:
07B1    ;
07B1    ;       R0 low bit set -> successful return
07B1    ;       R0 low bit clear -> invalid item identifier
07B1    ;
07B1    ; SIDE EFFECTS:
07B1    ;
07B1    ;       none
07B1    ;--
```

```
                              07B1  ;
                              07B1  ; This table is used to convert the pre V4 GETSYI item-codes to the
                              07B1  ; new ones, which have a different form.
                              07B1  ;
                              07B1  ; Old form:                          +---------+---------+
                              07B1  ;                                    ; 8 bits  ; 8 bits  ;
                              07B1  ;                                    +---------+---------+
                              07B1  ; SYI$_OLDVERSION =                       01        00
                              07B1  ; SYI$_OLDCPU =                           02        00
                              07B1  ; SYI$_OLDSID =                           02        01
                              07B1  ;
                              07B1  ; New form:                          +---+---------------+
                              07B1  ;                                    ; 4 ; 12 bits       ;
                              07B1  ;                                    +---+---------------+
                              07B1  ; compatibile with old =    0
                              07B1  ; EXE items =               1
                              07B1  ; FLD items =               2
                              07B1  ;
                              07B1
                              07B1  COMPAT:
               1000 0100      07B1          .WORD   SYI$_OLDVERSION,        SYI$_VERSION
               2000 0200      07B5          .WORD   SYI$_OLDCPU,            SYI$_CPU
               1001 0201      07B9          .WORD   SYI$_OLDSID,            SYI$_SID
                              07BD
                              07BD          .ENABLE LOCAL_BLOCK
                              07BD
                              07BD  CHECKITEM:
                     50   D4  07BD          CLRL    R0                      ; assume error
         51    F000  8F   B3  07BF          BITW    #^XF000, R1             ; is it a new item-code?
                     18   12  07C4          BNEQU   10$                     ; NEQU means it is
                 53  03   9A  07C6          MOVZBL  #3, R3                  ; setup to scan table
              52   E3 AF  3E  07C9          MOVAW   COMPAT-2, R2
                     82   B5  07CD  5$:     TSTW    (R2)+                   ; skip past new item-code
                 51  82   B1  07CF          CMPW    (R2)+, R1               ; does it match this old item-codes?
                     05   12  07D2          BNEQU   7$                      ; NEQU means it does not
                 51  62   B0  07D4          MOVW    (R2), R1                ; match, use the new itemcode instead
                     05   11  07D7          BRB     10$                     ; continue like nothing happened
                 F1  53   F5  07D9  7$:     SOBGTR  R3, 5$                  ; cycle through the table
                     55   11  07DC          BRB     900$                    ; error if it wasn't in the table
  52     51    04    0C   EF  07DE  10$:    EXTZV   #12,#4,R1,R2            ; get the structure number
  53     51    0C    00   EF  07E3          EXTZV   #0,#12,R1,R3            ; get the item number
               02    52   91  07E8          CMPB    R2, #MAXSTRUC           ; is it a legal structure number?
                     46   1A  07EB          BGTRU   900$                    ; GTRU means it is not
       F808 CF42     53   B1  07ED          CMPW    R3, MAXCOUNT-2[R2]      ; is it a legal item number?
                     3E   1A  07F3          BGTRU   900$                    ; GTRU means it is not
                              07F5          CASE    R2,<EXE$,FLD$>B,#1      ; goto the appropriate code
                              07FD
                 53  05   C4  07FD  EXE$:   MULL    #5, R3                  ; calc total offset
         53  F7FF CF43    9E  0800          MOVAB   EXETBL[R3], R3          ; get address of table element
                     1A   11  0806          BRB     50$
                              0808
                 53  07   C4  0808  FLD$:   MULL    #7, R3                  ; calc total offset
         53  FCFE CF43    9E  080B          MOVAB   FLDTBL[R3], R3          ; get address of table element
E0 AD    63    05    0B   EF  0811          EXTZV   #11,#5,(R3),BITSIZ(FP)  ; get (bitsiz-1) value
               E0    AD   D6  0817          INCL    BITSIZ(FP)              ; restore its original value
E4 AD    63    0B    00   EF  081A          EXTZV   #0,#11,(R3),BITPOS(FP)  ; get bitpos value
                     83   B5  0820          TSTW    (R3)+                   ; point to next longword
```

```
                              0822
            54    83    D0    0822   50$:     MOVL    (R3)+, R4                     ; get source address
                  0D    13    0825            BEQLU   100$                          ; NULL SOURCE MEANS RETIRED ITEM-CODE!!
                              0827                                                  ; IT ALSO MEANS PR$ KSP WILL NEVER BE
                              0827                                                  ; ABLE TO BE AN ITEM-CODE!
55    63    03    05    EF    0827            EXTZV   #5,#3,(R3),R5                 ; get DTYPE
53    63    05    00    EF    082C            EXTZV   #0,#5,(R3),R3                 ; get OUTLEN
                  50    D6    0831   70$:     INCL    R0                            ; success!
                              0833
                        05    0833   900$:    RSB                                   ; return to caller
                              0834
         FC AD    08    C8    0834   100$:    BISL    #<1@SYI_V_RETIRED>,FLAGS(FP) ; mark it as obsolete
            53    04    D0    0838            MOVL    #4,R3                         ; src length
      54    EC    AD    DE    083B            MOVAL   SPECIAL_SPACE(FP),R4          ; scratch area
                  64    D4    083F            CLRL    (R4)                          ; null answer now
            55    00    D0    0841            MOVL    #VALUE,R5                     ; dtype
                  EB    11    0844            BRB     70$                           ; success exit
                              0846
                              0846            .DISABLE LOCAL_BLOCK
```

```
0846          .SBTTL  PUTDATA - Put requested data in user buffer
0846
0846   ;++
0846   ;
0846   ; FUNCTIONAL DESCRIPTION:
0846   ;
0846   ;       This routine moves the requested data to the user's buffer and
0846   ;       returns the actual data length to the user. It assumes that the
0846   ;       user's buffer has been probed.
0846   ;
0846   ; CALLING SEQUENCE:
0846   ;
0846   ;       JSB/BSB
0846   ;
0846   ; INPUTS:
0846   ;
0846   ;       R1 = item identifier
0846   ;       R2 = data structure number
0846   ;       R3 = item length
0846   ;       R4 = item address
0846   ;       R5 = item type code
0846   ;       R6 = user buffer length
0846   ;       R7 = user buffer address
0846   ;       R8 = address to return length
0846   ;       BITSIZ(FP)
0846   ;       BITPOS(FP)
0846   ;
0846   ; IMPLICIT INPUTS:
0846   ;
0846   ;       none
0846   ;
0846   ; OUTPUTS:
0846   ;
0846   ;       none
0846   ;
0846   ; IMPLICIT OUTPUTS:
0846   ;
0846   ;       none
0846   ;
0846   ; ROUTINE VALUE:
0846   ;
0846   ;       R0 low bit set -> success
0846   ;       R0 low bit clear -> access violation on write of length
0846   ;
0846   ; SIDE EFFECTS:
0846   ;
0846   ;       Registers R1-R4 destroyed
0846   ;--
```

```
                            0846
                            0846  PUTDATA:
                            0846
                            0846  ;
                            0846  ; Call routine to check for special conditions
                            0846  ;
                            0846
      27 FC AD    03    E0  0846         BBS     #SYI_V_RETIRED,FLAGS(FP),15$ ; skip alot for oldie items
                  53    10  084B         BSBB    CHECK_SPC                    ; handle special items
            4F    50    E9  084D         BLBC    R0,50$                       ; were we successful?
                            0850
                            0850  ;
                            0850  ; Check for counted string, and find actual length if so.
                            0850  ;
                            0850
            55    02    D1  0850         CMPL    #CSTRING,R5                  ; is this special string?
                  03    12  0853         BNEQ    10$                          ; branch if not
            53    84    9A  0855         MOVZBL  (R4)+,R3                     ; get length and skip length byte
                            0858  ;
                            0858  ; Fetch the bitpos and bitsiz if it's a FLD item
                            0858  ;
            02    52    D1  0858  10$:   CMPL    R2, #SYI$C_FLDTYPE           ; is it FLD?
                  15    12  085B         BNEQU   15$                          ; NEQU means it is not
         50    E0 AD    D0  085D         MOVL    BITSIZ(FP), R0               ; get bitsiz
         51    E4 AD    D0  0861         MOVL    BITPOS(FP), R1               ; get bitpos
         E8 AD    64    D0  0865         MOVL    (R4), TEMPORARY(FP)          ; get copy of cell
         54    E8 AD    DE  0869         MOVAL   TEMPORARY(FP), R4            ; point to copy
  64    64    50    51    EF  086D         EXTZV   R1,R0,(R4),(R4)              ; fetch sub-field and save
                            0872  ;
                            0872  ; Move the data
                            0872  ;
                  28    BB  0872  15$:   PUSHR   #^M<R3,R5>                   ; save needed registers from movc
67   56   00   64   53    2C  0874         MOVC5   R3,(R4),#0,R6,(R7)           ; move data to user's buffer, zero fill
                  28    BA  087A         POPR    #^M<R3,R5>                   ; restore registers
                  58    D5  087C         TSTL    R8                           ; did caller want return length?
                  18    13  087E         BEQL    30$                          ; branch if not
                            0880         IFNOWRT #2,(R8),40$                  ; exit if word not writable
      02 FC AD    03    E1  0886         BBC     #SYI_V_RETIRED,FLAGS(FP),18$
                  53    D4  088B         CLRL    R3                           ; make his retlen null
            56    53    D1  088D  18$:   CMPL    R3,R6                        ; see how much was moved
                  03    15  0890         BLEQ    20$                          ; use valid data length if it fit
            53    56    D0  0892         MOVL    R6,R3                        ; else give him "too short" buffer size
            68    53    B0  0895  20$:   MOVW    R3,(R8)                      ; return length to user
                            0898
            50    01    3C  0898  30$:   MOVZWL  #SS$_NORMAL,R0               ; set success code
                        05  089B         RSB
                            089C
            50    0C    3C  089C  40$:   MOVZWL  #SS$_ACCVIO,R0               ; set error code
                        05  089F  50$:   RSB                                  ; return
```

```
08A0          .SBTTL  SPECIAL - Handle special conditions
08A0
08A0    ;++
08A0    ;
08A0    ; FUNCTIONAL DESCRIPTION:
08A0    ;
08A0    ;       These routines handle data items which must be transformed
08A0    ;       before they are returned to the user.  Generally, some
08A0    ;       transformation is applied to the data item and the newly
08A0    ;       computed item is stored in SPECIAL_SPACE on the stack.
08A0    ;       The handling routine then changes R4 to point to SPECIAL_SPACE
08A0    ;       so that PUTDATA will move the item from local storage.
08A0    ;
08A0    ; CALLING SEQUENCE:
08A0    ;
08A0    ;       JSB/BSB
08A0    ;
08A0    ; INPUTS:
08A0    ;
08A0    ;       R1 = item identifier
08A0    ;       R3 = item length
08A0    ;       R4 = item address/offset
08A0    ;       R9 = target CSB address
08A0    ;       R11 = target CSID
08A0    ;
08A0    ; IMPLICIT INPUTS:
08A0    ;
08A0    ;       none
08A0    ;
08A0    ; OUTPUTS:
08A0    ;
08A0    ;       none
08A0    ;
08A0    ; IMPLICIT OUTPUTS:
08A0    ;
08A0    ;       none
08A0    ;
08A0    ; ROUTINE VALUE:
08A0    ;
08A0    ;       none
08A0    ;
08A0    ; SIDE EFFECTS:
08A0    ;
08A0    ;       none
08A0    ;--
```

```
SYSG
Symb

EXE_
FETC
FLAG
FLD$
FLDT
GETS
GET_
GOTC
GOTN
GRET
IOC$
IOC$
IOC$
IOC$
IOC$
IOC$
IOC$
IOSB
IPL$
IPL$
ITML
LCK$
LCK$
LCK$
LCK$
LCK$
LNM$
LNM$
LOCA
LOCA
LOCK
LOCK
MAXC
MAXS
MAX_
MAX_
MMG$
MMG$
MMG$
MMG$
MPW$
MPW$
MPW$
MPW$
MPW$
MPW$
NAMC
NODE
NODE
NONE
NULL
NULL
OUTL
PCB$
PCB$
PFL$
PFL$
```

```
                         08A0
                         08A0  CHECK_SPC:
                         08A0
                         08A0  ;
                         08A0  ; Registers R5 - R8 are saved at this level and may be used by
                         08A0  ; the action routines without being saved.  Action routines are JSB'ed
                         08A0  ; to with R5 containing the address of SPECIAL_SPACE on the stack.
                         08A0  ;
                         08A0
        01E0 8F     BB   08A0          PUSHR   #^M<R5,R6,R7,R8>          ; save resisters
     55   EC AD     DE   08A4          MOVAL   SPECIAL_SPACE(FP),R5      ; local storage for action routine
             65     7C   08A8          CLRQ    (R5)                      ; clear the special buffer
          08 A5     7C   08AA          CLRQ    8(R5)
       50   01      3C   08AD          MOVZWL  #SS$_NORMAL,R0            ; assume success
       57   1A      D0   08B0          MOVL    #SPECIAL_LEN,R7           ; get number of table entries
   58   FDB4 CF     DE   08B3          MOVAL   SPECIAL,R8                ; get address of table
                         08B8  10$:
       88   51      B1   08B8          CMPW    R1,(R8)+                  ; does entry match item?
            13      13   08BB          BEQL    20$                       ; yes, go handle it
       58   04      C0   08BD          ADDL    #4,R8                     ; skip handler address
         F5 57      F5   08C0          SOBGTR  R7,10$                    ; scan rest of table
                         08C3
  03 FC AD  02      E1   08C3          BBC     #SYI_V_REMOTE_NODE,FLAGS(FP),35$ ; nonlocal noncluster info?
       54   55      D0   08C8          MOVL    R5,R4                     ; make the returned data null
     01E0 8F        BA   08CB  35$:    POPR    #^M<R5,R6,R7,R8>          ; restore registers
                    05   08CF          RSB
                         08D0
                         08D0  20$:
            98      16   08D0          JSB     @(R8)+                    ; call action routine
            F7      11   08D2          BRB     35$
                         08D4
                         08D4  ;
                         08D4  ; Data handling routines
                         08D4  ;**********************************************************************
                         08D4  ; ALL NON-CLUSTER SPECIAL DATA ITEMS SHOULD TEST REMOTE_NODE AS BELOW
                         08D4  ;**********************************************************************
                         08D4  ;
                         08D4
                         08D4  ;
                         08D4  ; Is the SCS code loaded?
                         08D4  ;
                         08D4
                         08D4  SPC_EXISTS:
  5C FC AD  02      E0   08D4          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
   00000000'EF      D5   08D9          TSTL    SCS$GA_EXISTS             ; is the cell empty?
            54      13   08DF          BEQLU   POINT_R4                  ; null cell means it doesn't exist
            65      D6   08E1          INCL    (R5)                      ; make result TRUE
            50      11   08E3          BRB     POINT_R4
                         08E5
                         08E5  ;
                         08E5  ; Processor registers require special instructions to fetch
                         08E5  ;
                         08E5
                         08E5  SPC_PROCREG:
  4B FC AD  02      E0   08E5          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
       65   54      DB   08EA          MFPR    R4,(R5)                   ; get the register contents
            46      11   08ED          BRB     POINT_R4
```

```
                          08EF
                          08EF   ;
                          08EF   ; Cell is stored as a negative, reverse it and return
                          08EF   ;
                          08EF
                          08EF   SPC_NEGATIVE:
      41 FC AD    02   E0 08EF           BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
            65    64   CE 08F4           MNEGL   (R4), (R5)                   ; Negate it
                  3C   11 08F7           BRB     POINT_R4
                          08F9
                          08F9   ;
                          08F9   ; This cluster item makes sense even when not in a cluster
                          08F9   ;
                          08F9
                          08F9   SPC_MEMBER:
65   FC AD   01   01   EF 08F9           EXTZV   #SYI_V_INCLUSTER,#1,FLAGS(FP),(R5) ; get the flag
                  34   11 08FF           BRB     POINT_R4
                          0901
                          0901   ;
                          0901   ; These are cluster only items
                          0901   ;
                          0901
                          0901   SPC_CLUB:
      2F FC AD    01   E1 0901           BBC     #SYI_V_INCLUSTER,FLAGS(FP),POINT_R4 ; return null if no cluster
    57   00000000'EF   D0 0906           MOVL    CLU$GL_CLUB,R7               ; get the address of the CLUB
            54  6744   9E 090D           MOVAB   (R7)[R4],R4                  ; get the address of the field
                  44   11 0911           BRB     FETCH_CLU
                          0913
                          0913   SPC_CSB:
      1D FC AD    01   E1 0913           BBC     #SYI_V_INCLUSTER,FLAGS(FP),POINT_R4 ; return null if no cluster
             00000918     0918   SPC_LOCK = .
                          0918           SETIPL  LOCK                         ; lock the database
                  43   10 091F           BSBB    VERIFY_CSB                   ; double check the CSB address
               11 50   E9 0921           BLBC    R0,POINT_R4                  ; exit if bad
            54  6944   9E 0924           MOVAB   (R9)[R4],R4                  ; get actual address of field
                  2D   11 0928           BRB     FETCH_CLU
                          092A
                          092A   LOCAL_SB:
                  51   DD 092A           PUSHL   R1                           ; save R1
    51   00000000'GF   DE 092C           MOVAL   G^SCS$GA_LOCALSB,R1          ; point to our local system block
                  1B   11 0933           BRB     GET_SB_FLD                   ; go get desired information from SB
                          0935
                          0935   POINT_R4:
            54    55   D0 0935           MOVL    R5,R4                        ; make R4 point to data
                       05 0938           RSB
                          0939
                          0939   SPC_SB:
      EC FC AD    01   E1 0939           BBC     #SYI_V_INCLUSTER,FLAGS(FP),LOCAL_SB ; use local SB if no cluster
                          093E           SETIPL  LOCK                         ; lock the database
                  1D   10 0945           BSBB    VERIFY_CSB                   ; double check the CSB address
               EB 50   E9 0947           BLBC    R0,POINT_R4                  ; exit if bad
                  51   DD 094A           PUSHL   R1                           ; save R1
         51    68 A9   D0 094C           MOVL    CSB$L_SB(R9),R1              ; get SB address
                          0950   GET_SB_FLD:
            54  6144   9E 0950           MOVAB   (R1)[R4],R4                  ; get actual address of field
                 51  8ED0 0954           POPL    R1                           ; restore R1
                          0957
```

```
                                  0957  FETCH_CLU:
                      3F    BB    0957          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; save the register past the MOVC3
          65    64    53    28    0959          MOVC3   R3,(R4),(R5)            ; get the data into the special buffer
                      3F    BA    095D          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; restore the registers
                                  095F          SETIPL  #0                      ; drop IPL again
                      D1    11    0962          BRB     POINT_R4
                                  0964
                                  0964  VERIFY_CSB:
                56    5B    32    0964          CVTWL   R11,R6                  ; get the system index
      56    00000000'FF46   D0    0967          MOVL    @CLU$GL_CLUSVEC[R6],R6  ; get the CSB address
                      0F    18    096F          BGEQ    99$                     ; GEQ means it is now unused
                59    56    D1    0971          CMPL    R6,R9                   ; is it the same as ours?
                      0A    12    0974          BNEQ    99$                     ; NEQ means it changed
           5B    4C   A6    D1    0976          CMPL    CSB$L_CSID(R6),R11      ; is the CSID the same?
                      04    12    097A          BNEQ    99$                     ; NEQ means it changed
                50    01    3C    097C          MOVZWL  #SS$_NORMAL,R0
                            05    097F          RSB
                                  0980
                                  0980  99$:    SETIPL  #0                      ; drop IPL
           FC AD      02    CA    0983          BICL2   #<1@SYI_V_INCLUSTER>,FLAGS(FP) ; reset the cluster flag
          50    028C  8F    3C    0987          MOVZWL  #SS$_NOSUCHNODE,R0      ; declare an error
                            05    098C          RSB
                                  098D
                                  098D  LOCK:
                            08    098D          .BYTE   IPL$_SCS
                                  098E          ASSUME  <.-SPC_LOCK> LE 512
                                  098E  ;
                                  098E  ;
                                  098E  ; Return total sizes for all page or swap files
                                  098E  ;
                                  098E  ; Input bit mask in R4
                                  098E  ;       bit 0
                                  098E  ;               0 -> page file
                                  098E  ;               1 -> swap file
                                  098E  ;       bit 1
                                  098E  ;               0 -> total space
                                  098E  ;               1 -> free space
                                  098E  ;       bit 2
                                  098E  ;               1 -> keeps it from being null
                                  098E  ;
                                  098E  ;
                                  098E  SPC_PAGESWAP:
        A2 FC AD      02    E0    098E          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
                   10 54    E8    0993          BLBS    R4,10$                  ; swap file
      57    00000000'EF     3C    0996          MOVZWL  SGN$GW_SWPFILCT,R7      ; first page file slot
      58    00000000'EF     D0    099D          MOVL    MMG$GL_MAXPFIDX,R8      ; last one
                      0C    11    09A4          BRB     20$                     ; resume in common code
                                  09A6  ;
                57    01    D0    09A6  10$:    MOVL    #1,R7                   ; always the first swap file slot
      58    00000000'EF     3C    09A9          MOVZWL  SGN$GW_SWPFILCT,R8
                      58    D7    09B0          DECL    R8                      ; max index for swap files
                                  09B2  ;
                      65    D4    09B2  20$:    CLRL    (R5)                    ; initial count
                58    57    C2    09B4          SUBL    R7,R8                   ; slots to count - 1
                      32    19    09B7          BLSS    60$                     ; none
      57    00000000'FF47   DE    09B9          MOVAL   @MMG$GL_PAGSWPVC[R7],R7 ; first slot
                56    87    D0    09C1  30$:    MOVL    (R7)+,R6                ; address of PFL structure
```

```
              00000000'EF   56   D1  09C4  1064          CMPL    R6,MMG$GL_NULLPFL          ; is it in use
                            13   13  09CB  1065          BEQL    50$                        ; no
                0E 23 A6    00   E1  09CD  1066          BBC     #PFL$V_INITED,PFL$B_FLAGS(R6),50$ ; not inited
                   06 54    01   E0  09D2  1067          BBS     #1,R4,40$                  ; count free space
                   65    14 A6   C0  09D6  1068          ADDL    PFL$L_BITMAPSIZ(R6),(R5)  ; total size / 8
                            04   11  09DA  1069          BRB     50$
                   65    18 A6   C0  09DC  1070 40$:     ADDL    PFL$L_FREPAGCNT(R6),(R5)  ; total free pages
                         DE 58   F4  09E0  1071 50$:     SOBGEQ  R8,30$                     ; loop over all files
                                     09E3  1072 ;
                   04 54    01   E0  09E3  1073          BBS     #1,R4,60$                  ; free space - already page count
                65    65    03   78  09E7  1074          ASHL    #3,(R5),(R5)               ; convert byte count to page count
                          FF47   31  09EB  1075 60$:     BRW     POINT_R4                   ; join common exit code
                                     09EE  1076
                         09EE          .SBTTL  NAMCSID - Get specified node CSID
                         09EE    ;++
                         09EE    ;
                         09EE    ; FUNCTIONAL DESCRIPTION:
                         09EE    ;
                         09EE    ;       Routine to convert a node name to a CSID. If a
                         09EE    ;       valid CSID or node name is specified, the standard conversion
                         09EE    ;       routine EXE$NAMCSID is simply called.  If, however, a CSID that implies
                         09EE    ;       a "wildcard" CSID (-1) is specified, then the next active node is
                         09EE    ;       chosen as the node CSID to pass to EXE$NAMCSID.  EXE$NAMCSID then
                         09EE    ;       returns the node's CSB address.
                         09EE    ;
                         09EE    ; INPUTS:
                         09EE    ;
                         09EE    ;       CSIDADR(AP) = address of specified CSID
                         09EE    ;       NODE(AP) = address of specified process name descriptor
                         09EE    ;
                         09EE    ; OUTPUTS:
                         09EE    ;
                         09EE    ;       R0 = success/failure of operation
                         09EE    ;       R4 = current process PCB address
                         09EE    ;       R9 = specified node CSB address
                         09EE    ;       R11 = specified node CSID
                         09EE    ;       @CSIDADR(AP) = specified node CSID or special "wildcard" context CSID
                         09EE    ;--
                         09EE
              00000008   09EE          CSIDADR  = 8
              0000000C   09EE          NODE     = 12
                         09EE
                         09EE    NAMCSID:
                         09EE          .ENABLE LOCAL_BLOCK
                         09EE
                         09EE    ;
                         09EE    ; MAKE SURE WE'RE IN A CLUSTER HERE
                         09EE    ;
56   00000000'EF    D0   09EE          MOVL    CLU$GL_CLUB,R6              ; GET CLUB ADDRESS
              09    13   09F5          BEQL    1$                          ; IF EQL, NOT IN CLUSTER
   04 1C A6   00    E1   09F7          BBC     #CLUB$V_CLUSTER,CLUB$L_FLAGS(R6),1$ ; IF CLEAR, NOT A CLUSTER
      FC AD   02    C8   09FC          BISL2   #<1@SYI_V_INCLUSTER>,FLAGS(FP) ; mark that we're in a cluster
                         0A00
     56   08 AC     D0   0A00    1$:   MOVL    CSIDADR(AP),R6              ; get CSID address
              4D    13   0A04          BEQL    19$                         ; if eql - none
                         0A06          IFWRT   #4,(R6),2$                  ; check access to CSID
            00CE    31   0A0C          BRW     50$
      50      66    D0   0A0F    2$:   MOVL    (R6),R0                     ; get CSID
              3F    13   0A12          BEQL    19$                         ; if eql - none
              79    14   0A14          BGTR    20$                         ; if gtr - standard CSID
                         0A16    ;
                         0A16    ; "Wildcard" type CSID specified
                         0A16    ;
                         0A16
   03 FC AD   01    E0   0A16          BBS     #SYI_V_INCLUSTER,FLAGS(FP),5$ ; are we in a cluster?
            00C4    31   0A1B          BRW     60$                         ; wildcarding without a cluster!
                         0A1E    5$:   SETIPL  80$                         ; lock the cluster database
      55      50    32   0A25          CVTWL   R0,R5                       ; get NIX (Node IndeX) from CSID
      FC AD   01    C8   0A28          BISL2   #<1@SYI_V_WILD>,FLAGS(FP)   ; mark wildcarding in effect
              55    B6   0A2C    10$:  INCW    R5                          ; increment NIX
     00000000'EF   55    B1  0A2E          CMPW    R5,CLU$GW_MAXINDEX           ; is NIX in valid range?
              03   1F  0A35                BLSSU   11$                          ; if LSSU, yep
            00A8   31  0A37                BRW     60$                          ; no more nodes
          55  55   3C  0A3A    11$:        MOVZWL  R5,R5                        ; clear out the top half of R5
 50    00000000'FF45   D0  0A3D            MOVL    @CLU$GL_CLUSVEC[R5],R0       ; get CSB address
              E5   18  0A45                BGEQ    10$                          ; if GEQ, unused - try next one
       7E    4C  A0    D0  0A47            MOVL    CSB$L_CSID(R0),-(SP)         ; get the CSID
                       0A4B                SETIPL  #0                           ; lower IPL to touch the argument list
          66    8E     D0  0A4E            MOVL    (SP)+,(R6)                   ; store CSID in argument list
                3C     11  0A51            BRB     20$
                           0A53
                           0A53  ;
                           0A53  ;         At this point, the CSID argument was defaulted
                           0A53  ;
    3C FC AD    01     E0  0A53  19$:      BBS     #SYI_V_INCLUSTER,FLAGS(FP),21$ ; let EXE$NAMCSID do it
      53  0C    AC     D0  0A58            MOVL    NODE(AP),R3                  ; get the nodename argument
                71     13  0A5C            BEQL    75$                          ; it was defaulted too, skip everything
                           0A5E
                           0A5E  ;
                           0A5E  ;         At this point, we're not in a cluster, but a nodename was specified,
                           0A5E  ;         see if it's the local one, if so succeed.
                           0A5E  ;
                           0A5E            IFNORD  #8,(R3),50$                  ; probe the descriptor
          52    63     7D  0A64            MOVQ    (R3),R2                      ; get the nodename descriptor
          52    52     3C  0A67            MOVZWL  R2,R2                        ; is the length legal?
                7D     13  0A6A            BEQL    65$                          ; EQL means nope
          52    0F     B1  0A6C            CMPW    #15,R2                       ; is it too long?
                78     1F  0A6F            BLSSU   65$                          ; LSSU means too long
                           0A71            IFNORD  R2,(R3),50$                  ; probe the string
  51     00000000'GF   DE  0A77            MOVAL   G^SCS$GA_LOCALSB,R1          ; point at the local SB
      55    44  A1     9E  0A7E            MOVAB   SB$T_NODENAME(R1),R5         ; get address of nodename
          85    52     91  0A82            CMPB    R2,(R5)+                     ; is it the right length?
                62     12  0A85            BNEQ    65$                          ; NEQ means no
      65    63  52     29  0A87            CMPC3   R2,(R3),(R5)                 ; is it the same nodename?
                5C     12  0A8B            BNEQ    65$                          ; NEQ means this is NOT the one
                40     11  0A8D            BRB     75$                          ; It is the local nodename, exit
                           0A8F
                           0A8F  ;
                           0A8F  ; Convert node name to CSID, if specified
                           0A8F  ;
    4E FC AD    01     E1  0A8F  20$:      BBC     #SYI_V_INCLUSTER,FLAGS(FP),60$ ; specified CSID   no cluster!
            5C  04     C0  0A94  21$:      ADDL    #4,AP                        ; make CSIDADR top argument
     00000000'EF       16  0A97            JSB     25$                          ; get into nonpaged code
                           0A9D            .SAVE_PSECT                          ; save current .PSECT context
                           0A9D
                           0A9D  ;  The reason for jumping to the nonpaged exec rather than dynamically
                           0A9D  ;  locking down pageable pages is that EXE$NAMCSID cannot be entered
                           0A9D  ;  above IPL 2 and the dynamic locking would cause that to happen. The
                           0A9D  ;  reason that EXE$NAMCSID must be entered at IPL 2 or lower is that it
                           0A9D  ;  touches the caller's argument list (which contains arguments that
                           0A9D  ;  could fault) and page faults are not allowed above IPL 2.
                           0A9D
                       00000000            .PSECT  AEXENONPAGED                 ; EXE$NAMCSID returns at IPL$_SYNCH
     00000AF1'EF       16  0000  25$:      JSB     EXE$NAMCSID                  ; get CSB address and CSID
                           0006            SETIPL  #0                           ; restore IPL - CSB is no longer locked
                       05  0009            RSB                                  ; go back to paged code
                           000A
```

```
                        D0000A9D        .RESTORE_PSECT                      ; get paged .PSECT context back
         5C      04  C2     0A9D        SUBL     #4,AP                      ; restore argument pointer
         5B      51  D0     0AA0        MOVL     R1,R11                     ; save CSID
   07 FC AD      00  E1     0AA3        BBC      #SYI_V_WILD,FLAGS(FP),30$ ; "wildcard" type CSID specified?
         66      51  B0     0AA8        MOVW     R1,(R6)                    ; restore node index context
      02 A6      01  AE     0AAB        MNEGW    #1,2(R6)                   ; set continuation context
                            0AAF   ;
                            0AAF   ; Check CSID address and return
                            0AAF   ;
         20      50  E9     0AAF   30$:  BLBC     R0,40$                    ; branch if error
         59      54  D0     0AB2        MOVL     R4,R9                      ; save CSB address
 54    00000000'EF  D0      0AB5        MOVL     CLU$GL_CLUB,R4             ; get address of Cluster Block
                 04  12     0ABC        BNEQU    32$                        ; NEQU means it's not null
                            0ABE        BUG_CHECK ICONCLUDAT,FATAL          ; oh oh
     54     10   A4  DE     0AC2   32$:  MOVAL    CLUB$L_LOCAL_CSB(R4),R4   ; get address of local CSB
         59      64  D1     0AC6        CMPL     (R4),R9                    ; see if local csb = target csb
                 04  13     0AC9        BEQL     75$                        ; EQL means target = local
      FC AD      04  C8     0ACB        BISL2    #<1@SYI_V_REMOTE_NODE>,FLAGS(FP) ; set the remote target flag
         50      01  3C     0ACF   75$:  MOVZWL   #SS$_NORMAL,R0            ; set success
 54    00000000'EF  D0      0AD2   40$:  MOVL     SCH$GL_CURPCB,R4          ; restore current PCB address
                            0AD9        SETIPL   #0                         ; make sure we can page fault
                     05     0ADC        RSB
                            0ADD
         50      0C  3C     0ADD   50$:  MOVZWL   #SS$_ACCVIO,R0            ; set access violation
                 F0  11     0AE0        BRB      40$                        ;
   50      0A00  8F  3C     0AE2   60$:  MOVZWL   #SS$_NOMORENODE,R0        ; set no more nodes
                 E9  11     0AE7        BRB      40$                        ;
   50      028C  8F  3C     0AE9   65$:  MOVZWL   #SS$_NOSUCHNODE,R0        ; set no such node
                 E2  11     0AEE        BRB      40$                        ;
                            0AF0
                     D8     0AF0   80$:  .BYTE    IPL$_SCS                  ; to lock the cluster database
                            0AF1        ASSUME   <.-5$> LE 512
                            0AF1
                            0AF1        .DISABLE LOCAL_BLOCK
```

```
                          0AF1          .SBTTL  EXE$NAMCSID - CONVERT NODE NAME TO CSID
                          0AF1    ;++
                          0AF1    ;
                          0AF1    ;       EXE$NAMCSID  -  CONVERT NODE NAME TO CSID
                          0AF1    ;
                          0AF1    ; FUNCTIONAL DESCRIPTION:
                          0AF1    ;       EXE$NAMCSID OBTAINS THE PROPER CSID AND CSB ADDRESS FOR A
                          0AF1    ;       STANDARD NODE SERVICE ARGUMENT LIST CONSISTING
                          0AF1    ;       OF A CSID/NODE-NAME PAIR.  THE ABSENCE OF BOTH SELECTS THE
                          0AF1    ;       CURRENT NODE.
                          0AF1    ;
                          0AF1    ;       NOTE THAT THE OPERATION OF THIS ROUTINE ONLY MAKES SENSE IN
                          0AF1    ;       A CLUSTER, THEREFORE A NOSUCHNODE ERROR WILL BE RETURNED IF
                          0AF1    ;       CLU$GL_CLUB = 0 ON ENTRY.
                          0AF1    ;
                          0AF1    ; CALLING SEQUENCE:
                          0AF1    ;       JSB/BSB EXE$NAMCSID
                          0AF1    ;
                          0AF1    ; INPUT PARAMETERS:
                          0AF1    ;       CSID(AP) - ADDRESS OF CSID SOURCE/DESTINATION (CSID)
                          0AF1    ;       NODENAME(AP) - POINTER TO NODE DESCRIPTOR TO CONVERT TO CSID
                          0AF1    ;
                          0AF1    ; IMPLICIT INPUTS:
                          0AF1    ;       @CLU$GL_CLUSVEC - VECTOR OF CSB ADDRESSES
                          0AF1    ;
                          0AF1    ; OUTPUT PARAMETERS:
                          0AF1    ;       R0 - COMPLETION STATUS
                          0AF1    ;       R1 - NODE IDENTIFICATION (CSID) OF NAMED NODE.
                          0AF1    ;       R4 - CSB ADDRESS OF NODE IF MATCH IS FOUND.
                          0AF1    ;       @CSID(AP) - NODE IDENTIFICATION (CSID) OF SELECTED NODE
                          0AF1    ;       IPL - IPL$_SYNCH (IPL UNCHANGED IF SS$_ACCVIO OR SS$_IVLOGNAM)
                          0AF1    ;
                          0AF1    ; COMPLETION CODES:
                          0AF1    ;       SS$_NORMAL   -  NORMAL SUCCESSFUL COMPLETION
                          0AF1    ;       SS$_IVLOGNAM  -  INVALID LOGICAL NAME STRING
                          0AF1    ;       SS$_NOSUCHNODE -  NONEXISTENT NODE OR INVALID CSID
                          0AF1    ;       SS$_ACCVIO  -  ACCESS VIOLATION FOR WRITE DESTINATION
                          0AF1    ;
                          0AF1    ; SIDE EFFECTS:
                          0AF1    ;       NONE
                          0AF1    ;--
                          0AF1
                00000004  0AF1          CSID     = 4                        ; special offset for EXE$NAMCSID
                00000008  0AF1          NODENAME = 8                        ; special offset for EXE$NAMCSID
                          0AF1
                          0AF1    EXE$NAMCSID:                              ; TRANSLATE PNAME TO CSID
                          0AF1          .ENABLE LOCAL_BLOCK
       50     12      DB  0AF1          MFPR    S^#PR$_IPL,R0               ; CHECK THE CURRENT IPL LEVEL
       50     02      D1  0AF4          CMPL    #IPL$_ASTDEL,R0             ; ARE WE ABOVE PAGE FAULT IPL?
              03      1E  0AF7          BGEQU   8$                          ; GOOD, WE CAN FAULT
            00B5      30  0AF9          BSBW    999$                        ; CANNOT BE CALLED ABOVE ASTDEL
 54    00000000'EF    D0  0AFC    8$:   MOVL    CLU$GL_CLUB,R4              ; GET THE CLUSTER BLOCK ADDRESS
              03      12  0B03          BNEQU   10$                         ; GOOD, WE'RE IN A CLUSTER
            00A3      31  0B05          BRW     NONEX                       ; CANNOT BE CALLED IF NOT IN A CLUSTER
      54    10 A4     D0  0B08    10$:  MOVL    CLUB$L_LOCAL_CSB(R4),R4     ; GET THE CSB ADDRESS
      50    04 AC     D0  0B0C          MOVL    CSID(AP),R0                 ; GET CSID ADDRESS
              1A      13  0B10          BEQL    30$                         ; NO CSID ADDRESS
                          0B12          IFWRT   #4,(R0),20$                 ; ERROR IF ACCESS VIOLATION
```

```
         50    0C   3C  0B18  35$:   MOVZWL  #SS$_ACCVIO,R0          ; SET ACCESS VIOLATION ERROR CODE
                    05  0B1B         RSB                             ; AND EXIT
         51    60   D0  0B1C  20$:   MOVL    (R0),R1                 ; NOW FETCH CSID
               0B   13  0B1F         BEQL    30$                     ; BRANCH IF NO CSID FOUND
               50   D4  0B21         CLRL    R0                      ; CLEAR CSID ADDRESS,
                        0B23                                         ; DON'T NEED TO REWRITE SAME VALUE
             009F   31  0B23         BRW     GOTCSID                 ; HAVE THE CSID, GO CHECK IT OUT
                        0B26
    50   0154 8F    3C  0B26  45$:   MOVZWL  #SS$_IVLOGNAM,R0        ; BAD NODENAME STRING
                    05  0B2B         RSB
                        0B2C
                        0B2C  ; NO CSID SPECIFIED (CSIDADR = 0 OR CSIDADR -> 0)
                        0B2C  ;
                        0B2C  ;      R4 -> LOCAL CSB
                        0B2C  ;      R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                        0B2C  ;      <R1,R2,R3> NOT INTERESTING
                        0B2C
       51   4C A4   D0  0B2C  30$:   MOVL    CSB$L_CSID(R4),R1       ; ASSUME LOCAL CSID
       53   08 AC   D0  0B30         MOVL    NODENAME(AP),R3         ; GET NODENAME ADDRESS IF SPECIFIED
               03   12  0B34         BNEQ    31$                     ; NEQ MEANS NAME WAS SPECIFIED
             008C   31  0B36         BRW     GOTCSID                 ; NO NAME SPECIFIED, USE CALLER'S CSID
                        0B39  31$:
                        0B39
                        0B39  ; MUST LOOK UP NODE NAME.  PROBE THE DESCRIPTOR AND THE STRING, AND THEN
                        0B39  ; COPY IT TO THE STACK SO THAT IT CAN BE ACCESSED AFTER WE RAISE IPL.
                        0B39  ;
                        0B39  ;      R4 -> CURRENT CSB
                        0B39  ;      R3 -> NODE NAME DESCRIPTOR (ACCESS NOT YET PROBED)
                        0B39  ;      R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                        0B39  ;      <R1,R2> NOT INTERESTING
                        0B39
                        0B39         IFNORD  #8,(R3),35$             ; PROBE THE DESCRIPTOR
         52    63   7D  0B3F         MOVQ    (R3),R2                 ; GET THE NODENAME DESCRIPTOR
         52    52   3C  0B42         MOVZWL  R2,R2                   ; IS THE LENGTH LEGAL?
               DF   13  0B45         BEQL    45$                     ; EQL MEANS NOPE
         52    0F   B1  0B47         CMPW    #15,R2                  ; IS IT TOO LONG?
               DA   1F  0B4A         BLSSU   45$                     ; LSSU MEANS TOO LONG
                        0B4C         IFNORD  R2,(R3),35$             ; PROBE THE STRING
         5E    10   C2  0B52         SUBL    #16,SP                  ; ALLOCATE BUFFER ON THE STACK
         51    5E   D0  0B55         MOVL    SP,R1                   ; TEMPORARY POINTER TO BUFFER
               52   DD  0B58         PUSHL   R2                      ; SAVE LENGTH OF NODE NAME STRING
         81    83   90  0B5A  40$:   MOVB    (R3)+,(R1)+             ; COPY NODE NAME STRING FROM USER'S
            FA 52   F5  0B5D         SOBGTR  R2,40$                  ;      BUFFER ONTO THE STACK
            52 8ED0     0B60         POPL    R2                      ; RESTORE LENGTH OF NODE NAME STRING
         53    5E   D0  0B63         MOVL    SP,R3                   ; POINTER TO NODE NAME BUFFER
               50   DD  0B66         PUSHL   R0                      ; SAVE THE CSIDADR ARGUMENT
  50  00000000'EF   3C  0B68         MOVZWL  CLU$GW_MAXINDEX,R0      ; GET THE NUMBER OF ENTRIES
               50   D7  0B6F         DECL    R0                      ; CONVERT TO HIGHEST OFFSET
                        0B71
                        0B71  ; SCAN CSB VECTOR TO LOOK FOR THIS NODE NAME
                        0B71  ;
                        0B71  ;      R4 -> CURRENT CSB
                        0B71  ;      R3 -> USER'S NODE NAME STRING  (IN BUFFER ON THE STACK)
                        0B71  ;      R2 =  USER'S NODE NAME LENGTH
                        0B71  ;      R0 =  COUNTER FOR CLUSVEC SLOTS
                        0B71
                        0B71  100$:  SETIPL  LOCKPAGE                ; LOCK DOWN THE REST OF THE ROUTINE
```

```
     51   00000000'FF40   D0  0B78  1341          MOVL    @CLU$GL_CLUSVEC[R0],R1   ; GET THE POINTER TO THE CSB
                     21   18  0B80  1342          BGEQ    155$                     ; GEQ MEANS UNUSED, TRY THE NEXT ONE
                     51   DD  0B82  1343          PUSHL   R1                       ; SAVE THE POINTER TO THE TARGET CSB
          51    68 A1     D0  0B84  1344          MOVL    CSB$L_SB(R1),R1          ; GET SB ADDRESS
                              0B88  1345
                              0B88  1346 ; IS THIS THE NODENAME?
                              0B88  1347 ;
                              0B88  1348 ;        R0 -   CLUSVEC INDEX
                              0B88  1349 ;        R1 ->  SB (SYSTEM BLOCK)
                              0B88  1350 ;        R2 ->  USER'S NODENAME LENGTH
                              0B88  1351 ;        R3 ->  USER'S NODENAME STRING
                              0B88  1352
                     55   DD  0B88  1353          PUSHL   R5                       ; SAVE R5
          55    44 A1     9E  0B8A  1354          MOVAB   SB$T_NODENAME(R1),R5     ; GET ADDRESS OF NODENAME
                85   52   91  0B8E  1355          CMPB    R2,(R5)+                 ; IS IT THE RIGHT LENGTH?
                     0A   12  0B91  1356          BNEQ    150$                     ; NEQ MEANS NO, TRY THE NEXT ONE
                     0F   BB  0B93  1357          PUSHR   #^M<R0,R1,R2,R3>         ; SAVE REGISTERS FOR THE CMPC3
          65    63   52   29  0B95  1358          CMPC3   R2,(R3),(R5)             ; IS IT THE SAME NODENAME?
                     0F   BA  0B99  1359          POPR    #^M<R0,R1,R2,R3>         ; RESTORE REGISTERS
                     18   13  0B9B  1360          BEQL    GOTNAM                   ; EQL MEANS THIS IS THE ONE
                              0B9D  1361
                              0B9D  1362 ; DID NOT FIND THE NODE BY NAME
                              0B9D  1363
                  55 8ED0     0B9D  1364 150$:    POPL    R5                       ; RESTORE R5
                  51 8ED0     0BA0  1365          POPL    R1                       ; RESTORE TARGET CSB ADDRESS
              CB  50   F4     0BA3  1366 155$:    SOBGEQ  R0,100$                  ; LOOP IF NOT DONE
                  8E   D5     0BA6  1367          TSTL    (SP)+                    ; THROW AWAY R0 FROM STACK
            5E    10   C0     0BA8  1368          ADDL    #16,SP                   ; POP NODE NAME BUFFER FROM STACK
                              0BAB  1369
                              0BAB  1370 ; EXIT WITH NONEXISTENT NODE STATUS
                              0BAB  1371
     50     028C 8F    3C     0BAB  1372 NONEX:   MOVZWL  #SS$_NOSUCHNODE,R0       ; SET ERROR STATUS
                       05     0BB0  1373          RSB                              ; AND RETURN TO CALLER
                              0BB1  1374
                              0BB1  1375 ; EXIT WITH A CRASH DUMP
                              0BB1  1376
                              0BB1  1377 999$:    BUG_CHECK  ICONCLUDAT,FATAL
                              0BB5  1378
                              0BB5  1379 ; FOUND THE NODE NAME, GET CSID FROM CSB AND CLEAN OFF THE STACK
                              0BB5  1380 ;
                              0BB5  1381 ;        R4 -> CURRENT CSB
                              0BB5  1382
                  55 8ED0     0BB5  1383 GOTNAM:  POPL    R5                       ; RESTORE R5
                  51 8ED0     0BB8  1384          POPL    R1                       ; RESTORE TARGET CSB ADDRESS
          51    4C A1  D0     0BBB  1385          MOVL    CSB$L_CSID(R1),R1        ; GET FULL CSID FOR NAME
                  50 8ED0     0BBF  1386          POPL    R0                       ; RESTORE CSIDADR ARGUMENT
            5E    10   C0     0BC2  1387          ADDL    #16,SP                   ; POP NODE NAME BUFFER FROM STACK
                              0BC5  1388
                              0BC5  1389 ; FOUND THE TARGET CSID, VERIFY IT
                              0BC5  1390 ;
                              0BC5  1391 ;        R4 -> CURRENT CSB
                              0BC5  1392 ;        R1 -> CSID OF TARGET NODE
                              0BC5  1393 ;        R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                              0BC5  1394 ;        <R2,R3> NOT INTERESTING
                              0BC5  1395
                              0BC5  1396 GOTCSID:
                              0BC5  1397          SETIPL  LOCKPAGE                 ; BLOCK SYSTEM EVENTS
```

```
                    52   51   3C  0BCC                    MOVZWL  R1,R2                   ; EXTRACT NODE INDEX
       00000000'EF   52   B1  0BCF                    CMPW    R2,CLU$GW_MAXINDEX      ; TEST AGAINST MAXIMUM VALUE
                    D3   1E  0BD6                    BGEQU   NONEX                   ; NONEXISTENT IF GEQU THAN MAXINDEX
52   00000000'FF42   D0  0BD8                    MOVL    @CLU$GL_CLUSVEC[R2],R2  ; GET CSB ADDRESS
                    C9   18  0BE0                    BGEQ    NONEX                   ; GEQ MEANS IT'S UNUSED
              4C A2   51   D1  0BE2                    CMPL    R1,CSB$L_CSID(R2)       ; CHECK FOR VALID CSID
                    C3   12  0BE6                    BNEQ    NONEX                   ; NOT THE SAME
                             0BE8
                             0BE8   RETURN:                                         ; SUCCESSFUL EXIT
                    54   52   D0  0BE8                    MOVL    R2,R4                   ; MOVE CSB ADDRESS OF TARGET
                             0BEB                                                   ; NORMAL STATUS EXIT
                         50   D5  0BEB                    TSTL    R0                      ; WAS CSID ADDRESS SPECIFIED
                         0A   13  0BED                    BEQL    910$                    ; NO, SKIP STORE OF CSID
                             0BEF                    SETIPL  #IPL$_ASTDEL            ; ALLOW PAGE FAULTS
                    60   51   D0  0BF2                    MOVL    R1,(R0)                 ; STORE CSID IN DESTINATION
                         50   D4  0BF5                    CLRL    R0                      ; DO NOT WRITE CSID A SECOND TIME
                         CC   11  0BF7                    BRB     GOTCSID                 ; MAKE SURE THAT CSID IS STILL VALID
                             0BF9
                    50   01   3C  0BF9   910$:    MOVZWL  #SS$_NORMAL,R0          ; SET SUCCESS STATUS
                              05  0BFC                    RSB                             ; AND RETURN TO CALLER
                             0BFD
                             0BFD   ; LOCK THIS PAGE DOWN WHEN WE RAISE IPL
                             0BFD   ;
                             0BFD
                             0BFD   LOCKPAGE:
                              08  0BFD                    .BYTE   IPL$_SCS                ; END OF LOCKED CODE REGION
                             0BFE                    ASSUME  <.-100$> LE 512
                             0BFE
                             0BFE                    .DISABLE LOCAL_BLOCK
                             0BFE
                             0BFE                    .END
```

```
SYSG
Symb

EXE$
EXE$
SS$_
SS$_
TIMA

PSEC
----
.  A
$ABS
YEXE

Phas
----
Init
Comm
Pass
Symb
Pass
Symb
Psec
Cros
Asse

The
1950
Ther
90 s
9 pa

Macr
----
_$25
-$25
TOTA

473

Ther

MACR
```

```
$$T1                         = 00000000                   CLUB$W_VOTES                  = 00000022
ACP$GB_BASEPRIO                *********   X    02        COMPAT                          000007B1 R      02
ACP$GB_DATACHK                 *********   X    02        CSB$L_CSID                    = 0000004C
ACP$GB_MAXREAD                 *********   X    02        CSB$L_SB                      = 00000068
ACP$GB_SWAPFLGS                *********   X    02        CSB$W_QUORUM                  = 00000052
ACP$GB_WINDOW                  *********   X    02        CSB$W_VOTES                   = 000[illegible]
ACP$GB_WRITBACK                *********   X    02        CSID                          = 00000004
ACP$GW_DINDXCACHE              *********   X    02        CSIDADR                       = 00000008
ACP$GW_DIRCACHE                *********   X    02        CSTRING                       = 00000002
ACP$GW_EXTCACHE                *********   X    02        CTL$GL_PCB                      *********   X    02
ACP$GW_EXTLIMIT                *********   X    02        EFN                           = 00000004
ACP$GW_FIDCACHE                *********   X    02        EXE$                            000007FD R      02
ACP$GW_HDRCACHE                *********   X    02        EXE$GETSYI                      00000000 RG     03
ACP$GW_MAPCACHE                *********   X    02        EXE$GL_ARCHFLAG                 *********   X    02
ACP$GW_QUOCACHE                *********   X    02        EXE$GL_CLITABL                  *********   X    02
ACP$GW_SYSACC                  *********   X    02        EXE$GL_DEFFLAGS                 *********   X    02
ACP$GW_WORKSET                 *********   X    02        EXE$GL_DYNAMIC_FLAGS            *********   X    02
ACP$V_READCHK                = 00000000  G                EXE$GL_LOCKRTRY                 *********   X    02
ACP$V_SWAPGRP                = 00000001  G                EXE$GL_MSGFLAGS                 *********   X    02
ACP$V_SWAPMAG                = 00000003  G                EXE$GL_RTIMESPT                 *********   X    02
ACP$V_SWAPPRV                = 00000002  G                EXE$GL_STATIC_FLAGS             *********   X    02
ACP$V_SWAPSYS                = 00000000  G                EXE$GL_SYSUIC                   *********   X    02
ACP$V_WRITECHK               = 00000001  G                EXE$GL_WSFLAGS                  *********   X    02
ARC$V_CHAR_EMUL              = 00000004                   EXE$GQ_BOOTTIME                 *********   X    02
ARC$V_DCML_EMUL              = 00000005                   EXE$NAMCSID                     00000AF1 R      02
ARC$V_DFLT_EMUL              = 00000008                   EXE$PROBEW                      *********   X    02
ARC$V_FFLT_EMUL              = 00000009                   EXE$V_BRK_DISUSER             = 00000003
ARC$V_GFLT_EMUL              = 0000000A                   EXE$V_BRK_TERM                = 00000002
ARC$V_HFLT_EMUL              = 0000000B                   EXE$V_BUGDUMP                   *********   X    02
ASTADR                       = 00000018                   EXE$V_BUGREBOOT                 *********   X    02
ASTPRM                       = 0000001C                   EXE$V_CJFLOAD                   *********   X    02
BIT...                       = 00000001                   EXE$V_CJFSYSRUJ                 *********   X    02
BITPOS                       = FFFFFFE4                   EXE$V_CLASS_PROT              = 00000000
BITSIZ                       = FFFFFFE0                   EXE$V_CONCEALED                 *********   X    02
BSTRING                      = 00000001                   EXE$V_CRDENABL                  *********   X    02
BUG$_ICONCLUDAT                *********   X    02        EXE$V_DISMOUMSG               = 00000001  G
CHECKITEM                      000007BD R       02        EXE$V_FATAL_BUG                 *********   X    02
CHECK_SPC                      000008A0 R       02        EXE$V_MOUNTMSG                = 00000000  G
CLU$GB_QDISK                   *********   X    02        EXE$V_MULTACP                   *********   X    02
CLU$GB_VAXCLUSTER              *********   X    02        EXE$V_NOAUTOCNF                 *********   X    02
CLU$GL_ALLOCLS                 *********   X    02        EXE$V_NOCLOCK                   *********   X    02
CLU$GL_CLUB                    *********   X    02        EXE$V_NOCLUSTER                 *********   X    02
CLU$GL_CLUSVEC                 *********   X    02        EXE$V_OPA0                    = 00000000  G
CLU$GW_LCKDIRWT                *********   X    02        EXE$V_POOLPGING                 *********   X    02
CLU$GW_MAXINDEX                *********   X    02        EXE$V_REBLDSYSD               = 00000001
CLU$GW_QDSKINTERVAL            *********   X    02        EXE$V_RESALLOC                  *********   X    02
CLU$GW_QDSKVOTES               *********   X    02        EXE$V_SAVEDUMP                  *********   X    02
CLU$GW_QUORUM                  *********   X    02        EXE$V_SBIERR                    *********   X    02
CLU$GW_RECNXINT                *********   X    02        EXE$V_SETTIME                   *********   X    02
CLU$GW_VOTES                   *********   X    02        EXE$V_SHRF11ACP                 *********   X    02
CLUB$B_FSYSID                = 00000026                   EXE$V_SSINHIBIT                 *********   X    02
CLUB$L_FLAGS                 = 0000001C                   EXE$V_SYSPAGING                 *********   X    02
CLUB$L_LOCAL_CSB             = 00000010                   EXE$V_SYSUAFALT                 *********   X    02
CLUB$Q_FTIME                 = 0000002C                   EXE$V_SYSWRTABL                 *********   X    02
CLUB$V_CLUSTER               = 00000000                   EXE$V_WRITESYSPARAMS          = 00000001
CLUB$W_NODES                 = 00000024                   EXE$V_XQP_RESIDENT            = 00000000
CLUB$W_QUORUM                = 00000020                   EXETBC                          00000004 R      02
```

```
EXE_GETSYI            000006D7 R       02    PFL$L_FREPAGCNT     = 00000018
FETCH_CLU             00000957 R       02    PFL$V_INITED        = 00000000
FLAGS               = FFFFFFFC               POINT_R4              00000935 R       02
FLD$                  00000808 R       02    PQL$GDASTLM           ********     X   02
FLDTBL                0000050E R       02    PQL$GDBIOLM           ********     X   02
GETSYISW            = 00000000               PQL$GDBYTLM           ********     X   02
GET_SB_FLD            00000950 R       02    PQL$GDCPULM           ********     X   02
GOTCSID               00000BC5 R       02    PQL$GDDIOLM           ********     X   02
GOTNAM                00000BB5 R       02    PQL$GDENQLM           ********     X   02
GRET                  0000076A R       02    PQL$GDFILLM           ********     X   02
IOC$GW_LAMAPREG       ********     X   02    PQL$GDJTQUOTA         ********     X   02
IOC$GW_MAXBUF         ********     X   02    PQL$GDPGFLQUOTA       ********     X   02
IOC$GW_MBXBFQUO       ********     X   02    PQL$GDPRCLM           ********     X   02
IOC$GW_MBXMXMSG       ********     X   02    PQL$GDTQELM           ********     X   02
IOC$GW_MBXNMMSG       ********     X   02    PQL$GDWSDEFAULT       ********     X   02
IOC$GW_MVTIMEOUT      ********     X   02    PQL$GDWSEXTENT        ********     X   02
IOC$GW_XFMXRATE       ********     X   02    PQL$GDWSQUOTA         ********     X   02
IOSB                = 00000014               PQL$GMASTLM           ********     X   02
IPL$_ASTDEL         = 00000002               PQL$GMBIOLM           ********     X   02
IPL$_SCS            = 00000008               PQL$GMBYTLM           ********     X   02
ITMLST              = 00000010               PQL$GMCPULM           ********     X   02
LCK$GL_EXTRASTK       ********     X   02    PQL$GMDIOLM           ********     X   02
LCK$GL_HTBLSIZ        ********     X   02    PQL$GMENQLM           ********     X   02
LCK$GL_IDTBLMAX       ********     X   02    PQL$GMFILLM           ********     X   02
LCK$GL_IDTBLSIZ       ********     X   02    PQL$GMJTQUOTA         ********     X   02
LCK$GL_WAITTIME       ********     X   02    PQL$GMPGFLQUOTA       ********     X   02
LNM$GL_HTBLSIZP       ********     X   02    PQL$GMPRCLM           ********     X   02
LNM$GL_HTBLSIZS       ********     X   02    PQL$GMTQELM           ********     X   02
LOCAL_SB              0000092A R       02    PQL$GMWSDEFAULT       ********     X   02
LOCAL_SPACE         = FFFFFFE0               PQL$GMWSEXTENT        ********     X   02
LOCK                  0000098D R       02    PQL$GMWSQUOTA         ********     X   02
LOCKPAGE              00000BFD R       02    PR$S_SID_TYPE       = 00000008
MAXCOUNT              00000000 R       02    PR$V_SID_TYPE       = 00000018
MAXSTRUC            = 00000002               PR$_IPL             = 00000012
MAX_EXE_ITEM        = 00000101               PR$_SID             = 0000003E
MAX_FLD_ITEM        = 0000002A               PSL$S_PRVMOD        = 00000002
MMG$GL_MAXPFIDX       ********     X   02    PSL$V_PRVMOD        = 00000016
MMG$GL_NULLPFL        ********     X   02    PUTDATA               00000846 R       02
MMG$GL_PAGSWPVC       ********     X   02    RETURN                00000BE8 R       02
MMG$GL_PHYPGCNT       ********     X   02    SB$B_HWVERS         = 00000038
MPW$GB_PRIO           ********     X   02    SB$B_SYSTEMID       = 00000018
MPW$GL_THRESH         ********     X   02    SB$Q_SWINCARN       = 0000002C
MPW$GL_WAITLIM        ********     X   02    SB$T_HWTYPE         = 00000034
MPW$GW_HILIM          ********     X   02    SB$T_NODENAME       = 00000044
MPW$GW_LOLIM          ********     X   02    SB$T_SWTYPE         = 00000024
MPW$GW_MPWPFC         ********     X   02    SB$T_SWVERS         = 00000028
NAMCSID               000009EE R       02    SCH$CLREF             ********     X   02
NODE                = 0000000C               SCH$GL_AWSTIME        ********     X   02
NODENAME            = 00000008               SCH$GL_BORROWLIM      ********     X   02
NONEX                 00000BAB R       02    SCH$GL_CURPCB         ********     X   02
NULLARG1            = 00000008               SCH$GL_GROWLIM        ********     X   02
NULLARG2            = 0000000C               SCH$GL_PFRATH         ********     X   02
OUTLEN              = 00000004               SCH$GL_PFRATL         ********     X   02
PCB$L_PID           = 00000060               SCH$GL_PFRATS         ********     X   02
PCB$W_ASTCNT        = 00000038               SCH$GL_SWPRATE        ********     X   02
PFL$B_FLAGS         = 00000023               SCH$GL_WSDEC          ********     X   02
PFL$L_BITMAPSIZ     = 00000014               SCH$GL_WSINC          ********     X   02
```

```
SCH$GW_AWSMIN                    ********  X   02     SGN$GL_NPAGEVIR                  ********  X   02
SCH$GW_DORMANTWAIT               ********  X   02     SGN$GL_PAGEDYN                   ********  X   02
SCH$GW_IOTA                      ********  X   02     SGN$GL_PE1                       ********  X   02
SCH$GW_LONGWAIT                  ********  X   02     SGN$GL_PE2                       ********  X   02
SCH$GW_QUAN                      ********  X   02     SGN$GL_PE3                       ********  X   02
SCH$GW_SWPFAIL                   ********  X   02     SGN$GL_PE4                       ********  X   02
SCH$POSTEF                       ********  X   02     SGN$GL_PE5                       ********  X   02
SCS$GA_EXISTS                    ********  X   02     SGN$GL_PE6                       ********  X   02
SCS$GA_LOCALSB                   ********  X   02     SGN$GL_SPTREQ                    ********  X   02
SCS$GB_NODENAME                  ********  X   02     SGN$GL_SRPCNT                    ********  X   02
SCS$GB_PAMXPORT                  ********  X   02     SGN$GL_SRPCNTV                   ********  X   02
SCS$GB_PANOPOLL                  ********  X   02     SGN$GL_SRPMIN                    ********  X   02
SCS$GB_PANPOLL                   ********  X   02     SGN$GL_SRPSIZE                   ********  X   02
SCS$GB_PASANITY                  ********  X   02     SGN$GL_USER3                     ********  X   02
SCS$GB_SYSTEMID                  ********  X   02     SGN$GL_USER4                     ********  X   02
SCS$GB_SYSTEMIDH                 ********  X   02     SGN$GL_USERD1                    ********  X   02
SCS$GB_UDABURST                  ********  X   02     SGN$GL_USERD2                    ********  X   02
SCS$GW_BDTCNT                    ********  X   02     SGN$GL_VMS5                      ********  X   02
SCS$GW_CDTCNT                    ********  X   02     SGN$GL_VMS6                      ********  X   02
SCS$GW_FLOWCUSH                  ********  X   02     SGN$GL_VMS7                      ********  X   02
SCS$GW_MAXDG                     ********  X   02     SGN$GL_VMS8                      ********  X   02
SCS$GW_MAXMSG                    ********  X   02     SGN$GL_VMSD1                     ********  X   02
SCS$GW_PAPOLINT                  ********  X   02     SGN$GL_VMSD2                     ********  X   02
SCS$GW_PAPOOLIN                  ********  X   02     SGN$GL_VMSD3                     ********  X   02
SCS$GW_PAPPDDG                   ********  X   02     SGN$GL_VMSD4                     ********  X   02
SCS$GW_PASTMOUT                  ********  X   02     SGN$GW_CTLIMGLIM                 ********  X   02
SCS$GW_PRCPOLINT                 ********  X   02     SGN$GW_CTLPAGES                  ********  X   02
SCS$GW_RDTCNT                    ********  X   02     SGN$GW_DFPFC                     ********  X   02
SGN$GB_KFILSTCT                  ********  X   02     SGN$GW_GBLSECNT                  ********  X   02
SGN$GB_PGTBPFC                   ********  X   02     SGN$GW_IMGIOCNT                  ********  X   02
SGN$GB_STARTUP_P1                ********  X   02     SGN$GW_ISPPGCT                   ********  X   02
SGN$GB_STARTUP_P2                ********  X   02     SGN$GW_MAXPRCCT                  ********  X   02
SGN$GB_STARTUP_P3                ********  X   02     SGN$GW_MAXPSTCT                  ********  X   02
SGN$GB_STARTUP_P4                ********  X   02     SGN$GW_MINWSCNT                  ********  X   02
SGN$GB_STARTUP_P5                ********  X   02     SGN$GW_PAGFILCT                  ********  X   02
SGN$GB_STARTUP_P6                ********  X   02     SGN$GW_PCHANCNT                  ********  X   02
SGN$GB_STARTUP_P7                ********  X   02     SGN$GW_PIOPAGES                  ********  X   02
SGN$GB_STARTUP_P8                ********  X   02     SGN$GW_PIXSCAN                   ********  X   02
SGN$GB_SYSPFC                    ********  X   02     SGN$GW_SWPFILCT                  ********  X   02
SGN$GB_TAILORED                  ********  X   02     SGN$GW_SWPFILES                  ********  X   02
SGN$GL_BALSETCT                  ********  X   02     SGN$GW_SYSDWSCT                  ********  X   02
SGN$GL_EXTRACPU                  ********  X   02     SGN$GW_TPWAIT                    ********  X   02
SGN$GL_EXUSRSTK                  ********  X   02     SGN$GW_WSLMXSKP                  ********  X   02
SGN$GL_FREEGOAL                  ********  X   02     SGN$V_LOADCHKPRT               = 00000001  G
SGN$GL_FREELIM                   ********  X   02     SGN$V_LOADERAPAT               = 00000000  G
SGN$GL_GBLPAGFIL                 ********  X   02     SGN$V_LOADMTACCESS             = 00000002  G
SGN$GL_IRPCNT                    ********  X   02     SIZ...                         = 00000001
SGN$GL_IRPCNTV                   ********  X   02     SPC_CLUB                         00000901 R     02
SGN$GL_LOADFLAGS                 ********  X   02     SPC_CSB                          00000913 R     02
SGN$GL_LRPCNT                    ********  X   02     SPC_EXISTS                       000008D4 R     02
SGN$GL_LRPCNTV                   ********  X   02     SPC_LOCK                       = 00000918 R     02
SGN$GL_LRPMIN                    ********  X   02     SPC_MEMBER                       000008F9 R     02
SGN$GL_LRPSIZE                   ********  X   02     SPC_NEGATIVE                     000008EF R     02
SGN$GL_MAXGPGCT                  ********  X   02     SPC_PAGESWAP                     0000098E R     02
SGN$GL_MAXVPGCT                  ********  X   02     SPC_PROCREG                      000008E5 R     02
SGN$GL_MAXWSCNT                  ********  X   02     SPC_SB                           00000939 R     02
SGN$GL_NPAGEDYN                  ********  X   02     SPECIAL                          0000063B R     02
```

```
SPECIAL_LEN                         = 0000001A              SYI$_CPU                            = 00002000
SPECIAL_SPACE                       = FFFFFFEC              SYI$_CRDENABLE                      = 00002002
SS$_ACCVIO                          = 0000000C              SYI$_CTLIMGLIM                      = 00001027
SS$_BADPARAM                        = 00000014              SYI$_CTLPAGES                       = 00001026
SS$_EXASTLM                         = 00002A04              SYI$_DEADLOCK_WAIT                  = 0000105E
SS$_IVLOGNAM                        = 00000154              SYI$_DECIMAL_EMULATED               = 0000201F
SS$_NOMORENODE                      = 00000A00              SYI$_DEFMBXBUFQUO                   = 00001050
SS$_NORMAL                          = 00000001              SYI$_DEFMBXMXMSG                    = 00001051
SS$_NOSUCHNODE                      = 0000028C              SYI$_DEFMBXNUMMSG                   = 00001052
STEP                                = 00000005              SYI$_DEFPRI                         = 000010B7
SWP$GB_PRIO                           ********   X    02    SYI$_DEFQUEPRI                      = 000010E2
SWP$GL_SWPPGCNT                       ********   X    02    SYI$_DISK_QUORUM                    = 000010DC
SWP$GW_SWPINC                         ********   X    02    SYI$_DISMOUMSG                      = 00002015
SYI$C_FLDTYPE                       = 00000002              SYI$_DLCKEXTRASTK                   = 00001011
SYI$_ACP_BASEPRIO                   = 000010B5              SYI$_DORMANTWAIT                    = 000010F1
SYI$_ACP_DATACHECK                  = 000010B4              SYI$_DUMPBUG                        = 00002003
SYI$_ACP_DINDXCACHE                 = 00001101              SYI$_D_FLOAT_EMULATED               = 00002020
SYI$_ACP_DIRCACHE                   = 000010AA              SYI$_EXTRACPU                       = 0000104C
SYI$_ACP_EXTCACHE                   = 000010AD              SYI$_EXUSRSTK                       = 0000101B
SYI$_ACP_EXTLIMIT                   = 000010AE              SYI$_FREEGOAL                       = 00001054
SYI$_ACP_FIDCACHE                   = 000010AC              SYI$_FREELIM                        = 00001053
SYI$_ACP_HDRCACHE                   = 000010A9              SYI$_F_FLOAT_EMULATED               = 00002021
SYI$_ACP_MAPCACHE                   = 000010A8              SYI$_GBLPAGES                       = 00001007
SYI$_ACP_MAXREAD                    = 000010B1              SYI$_GBLPAGFIL                      = 00001008
SYI$_ACP_MULTIPLE                   = 00002005              SYI$_GBLSECTIONS                    = 00001006
SYI$_ACP_QUOCACHE                   = 000010AF              SYI$_GROWLIM                        = 00001055
SYI$_ACP_REBLDSYSD                  = 00002029              SYI$_G_FLOAT_EMULATED               = 00002022
SYI$_ACP_SHARE                      = 0000200C              SYI$_H_FLOAT_EMULATED               = 00002023
SYI$_ACP_SWAPFLGS                   = 000010B6              SYI$_IJOBLIM                        = 000010B8
SYI$_ACP_SYSACC                     = 000010B0              SYI$_IMGIOCNT                       = 00001028
SYI$_ACP_WINDOW                     = 000010B2              SYI$_INTSTKPAGES                    = 00001010
SYI$_ACP_WORKSET                    = 000010AB              SYI$_IOTA                           = 0000103D
SYI$_ACP_WRITEBACK                  = 000010B3              SYI$_IRPCOUNT                       = 00001013
SYI$_ACP_XQP_RES                    = 00002025              SYI$_IRPCOUNTV                      = 00001014
SYI$_ALLOCLASS                      = 000010E5              SYI$_KFILSTCNT                      = 00001005
SYI$_ARCHFLAG                       = 000010DA              SYI$_LAMAPREGS                      = 00001059
SYI$_AWSMIN                         = 00001038              SYI$_LASTEXE                        = 00001102
SYI$_AWSTIME                        = 00001039              SYI$_LASTFLD                        = 0000202B
SYI$_BALSETCNT                      = 00001012              SYI$_LGI_BRK_DISUSER                = 00002028
SYI$_BJOBLIM                        = 000010B9              SYI$_LGI_BRK_LIM                    = 000010E8
SYI$_BOOTTIME                       = 000010BF              SYI$_LGI_BRK_TERM                   = 00002027
SYI$_BORROWLIM                      = 00001056              SYI$_LGI_BRK_TMO                    = 000010E9
SYI$_BUGCHECKFATAL                  = 00002004              SYI$_LGI_HID_TIM                    = 000010EA
SYI$_BUGREBOOT                      = 00002001              SYI$_LGI_PWD_TMO                    = 000010EB
SYI$_CHANNELCNT                     = 00001024              SYI$_LGI_RETRY_LIM                  = 000010E6
SYI$_CHARACTER_EMULATED             = 0000201E              SYI$_LGI_RETRY_TMO                  = 000010E7
SYI$_CJFLOAD                        = 00002019              SYI$_LNMPHASHTBL                    = 00001072
SYI$_CJFSYSRUJ                      = 0000201A              SYI$_LNMSHASHTBL                    = 00001071
SYI$_CLASS_PROT                     = 0000201D              SYI$_LOADCHKPRT                     = 00002017
SYI$_CLISYMTBL                      = 0000105B              SYI$_LOADERAPT                      = 00002016
SYI$_CLUSTER_FSYSID                 = 000010CD              SYI$_LOADMTACCESS                   = 00002024
SYI$_CLUSTER_FTIME                  = 000010CE              SYI$_LOCKDIRWT                      = 000010EF
SYI$_CLUSTER_MEMBER                 = 000010CF              SYI$_LOCKIDTBL                      = 0000105C
SYI$_CLUSTER_NODES                  = 000010CA              SYI$_LOCKIDTBL_MAX                  = 000010C0
SYI$_CLUSTER_QUORUM                 = 000010CC              SYI$_LOCKRETRY                      = 00001057
SYI$_CLUSTER_VOTES                  = 000010CB              SYI$_LONGWAIT                       = 0000103E
SYI$_CONCEAL_DEVICES                = 00002012              SYI$_LRPCOUNT                       = 0000101C
```

```
SYI$_LRPCOUNTV                   = 0000101D          SYI$_PFRATH                      = 00001034
SYI$_LRPMIN                      = 0000101F          SYI$_PFRATL                      = 00001033
SYI$_LRPSIZE                     = 0000101E          SYI$_PFRATS                      = 00001035
SYI$_MAXBUF                      = 0000104F          SYI$_PHYSICALPAGES               = 00001032
SYI$_MAXPROCESSCNT               = 00001009          SYI$_PIOPAGES                    = 00001025
SYI$_MAXQUEPRI                   = 000010E3          SYI$_PIXSCAN                     = 0000100A
SYI$_MAXSYSGROUP                 = 0000104D          SYI$_POOLPAGING                  = 00002009
SYI$_MINWSCNT                    = 0000100C          SYI$_PQL_DASTLM                  = 0000108E
SYI$_MOUNTMSG                    = 00002014          SYI$_PQL_DBIOLM                  = 00001090
SYI$_MPW_HILIMIT                 = 0000102B          SYI$_PQL_DBYTLM                  = 00001092
SYI$_MPW_LOLIMIT                 = 0000102C          SYI$_PQL_DCPULM                  = 00001094
SYI$_MPW_PRIO                    = 0000102D          SYI$_PQL_DDIOLM                  = 00001096
SYI$_MPW_THRESH                  = 0000102F          SYI$_PQL_DENQLM                  = 000010A6
SYI$_MPW_WAITLIMIT               = 00001030          SYI$_PQL_DFILLM                  = 00001098
SYI$_MPW_WRTCLUSTER              = 0000102A          SYI$_PQL_DJTQUOTA                = 000010EC
SYI$_MVTIMEOUT                   = 0000104E          SYI$_PQL_DPGFLQUOTA              = 0000109A
SYI$_NJOBLIM                     = 000010BA          SYI$_PQL_DPRCLM                  = 0000109C
SYI$_NOAUTOCONFIG                = 00002006          SYI$_PQL_DTQELM                  = 0000109E
SYI$_NOCLOCK                     = 00002007          SYI$_PQL_DWSDEFAULT              = 000010A0
SYI$_NOCLUSTER                   = 00002008          SYI$_PQL_DWSEXTENT               = 000010A4
SYI$_NODENAME                    = 000010D9          SYI$_PQL_DWSQUOTA                = 000010A2
SYI$_NODE_AREA                   = 0000201B          SYI$_PQL_MASTLM                  = 0000108F
SYI$_NODE_CSID                   = 000010D0          SYI$_PQL_MBIOLM                  = 00001091
SYI$_NODE_HWTYPE                 = 000010D7          SYI$_PQL_MBYTLM                  = 00001093
SYI$_NODE_HWVERS                 = 000010D8          SYI$_PQL_MCPULM                  = 00001095
SYI$_NODE_NUMBER                 = 0000201C          SYI$_PQL_MDIOLM                  = 00001097
SYI$_NODE_QUORUM                 = 000010D2          SYI$_PQL_MENQLM                  = 000010A7
SYI$_NODE_SWINCARN               = 000010D4          SYI$_PQL_MFILLM                  = 00001099
SYI$_NODE_SWTYPE                 = 000010D5          SYI$_PQL_MJTQUOTA                = 000010ED
SYI$_NODE_SWVERS                 = 000010D6          SYI$_PQL_MPGFLQUOTA              = 0000109B
SYI$_NODE_SYSTEMID               = 000010D3          SYI$_PQL_MPRCLM                  = 0000109D
SYI$_NODE_VOTES                  = 000010D1          SYI$_PQL_MTQELM                  = 0000109F
SYI$_NPAGEDYN                    = 00001016          SYI$_PQL_MWSDEFAULT              = 000010A1
SYI$_NPAGEVIR                    = 00001017          SYI$_PQL_MWSEXTENT               = 000010A5
SYI$_OLDCPU                      = 00000200          SYI$_PQL_MWSQUOTA                = 000010A3
SYI$_OLDSID                      = 00000201          SYI$_PRCPOLINTERVAL              = 00001069
SYI$_OLDVERSION                  = 00000100          SYI$_PROCSECTCNT                 = 0000100B
SYI$_PAGEDYN                     = 00001018          SYI$_QDSKINTERVAL                = 000010E4
SYI$_PAGEFILE_FREE               = 000010F4          SYI$_QDSKVOTES                   = 000010F0
SYI$_PAGEFILE_PAGE               = 000010F2          SYI$_QUANTUM                     = 00001029
SYI$_PAGFILCNT                   = 0000100D          SYI$_QUORUM                      = 000010BC
SYI$_PAGTBLPFC                   = 00001003          SYI$_REALTIME_SPTS               = 0000105A
SYI$_PAMAXPORT                   = 000010E0          SYI$_RECNXINTERVAL               = 000010BE
SYI$_PANOPOLL                    = 000010F8          SYI$_RESALLOC                    = 00002010
SYI$_PANUMPOLL                   = 0000106C          SYI$_RESHASHTBL                  = 0000105D
SYI$_PAPOLLINTERVAL              = 0000106D          SYI$_RJOBLIM                     = 000010BB
SYI$_PAPOOLINTERVAL              = 0000106E          SYI$_RMS_DFMBC                   = 00001084
SYI$_PASANITY                    = 000010E1          SYI$_RMS_DFMBFHSH                = 0000108A
SYI$_PASTDGBUF                   = 0000106B          SYI$_RMS_DFMBFIDX                = 00001089
SYI$_PASTIMOUT                   = 0000106A          SYI$_RMS_DFMBFREL                = 00001088
SYI$_PE1                         = 000010F9          SYI$_RMS_DFMBFSDK                = 00001085
SYI$_PE2                         = 000010FA          SYI$_RMS_DFMBFSMT                = 00001086
SYI$_PE3                         = 000010FB          SYI$_RMS_DFMBFSUR                = 00001087
SYI$_PE4                         = 000010FC          SYI$_RMS_DFNBC                   = 00001100
SYI$_PE5                         = 000010FD          SYI$_RMS_EXTEND_SIZE             = 0000108C
SYI$_PE6                         = 000010FE          SYI$_RMS_FILEPROT                = 0000108D
SYI$_PFCDEFAULT                  = 00001002          SYI$_RMS_GBLBUFQUO               = 000010FF
```

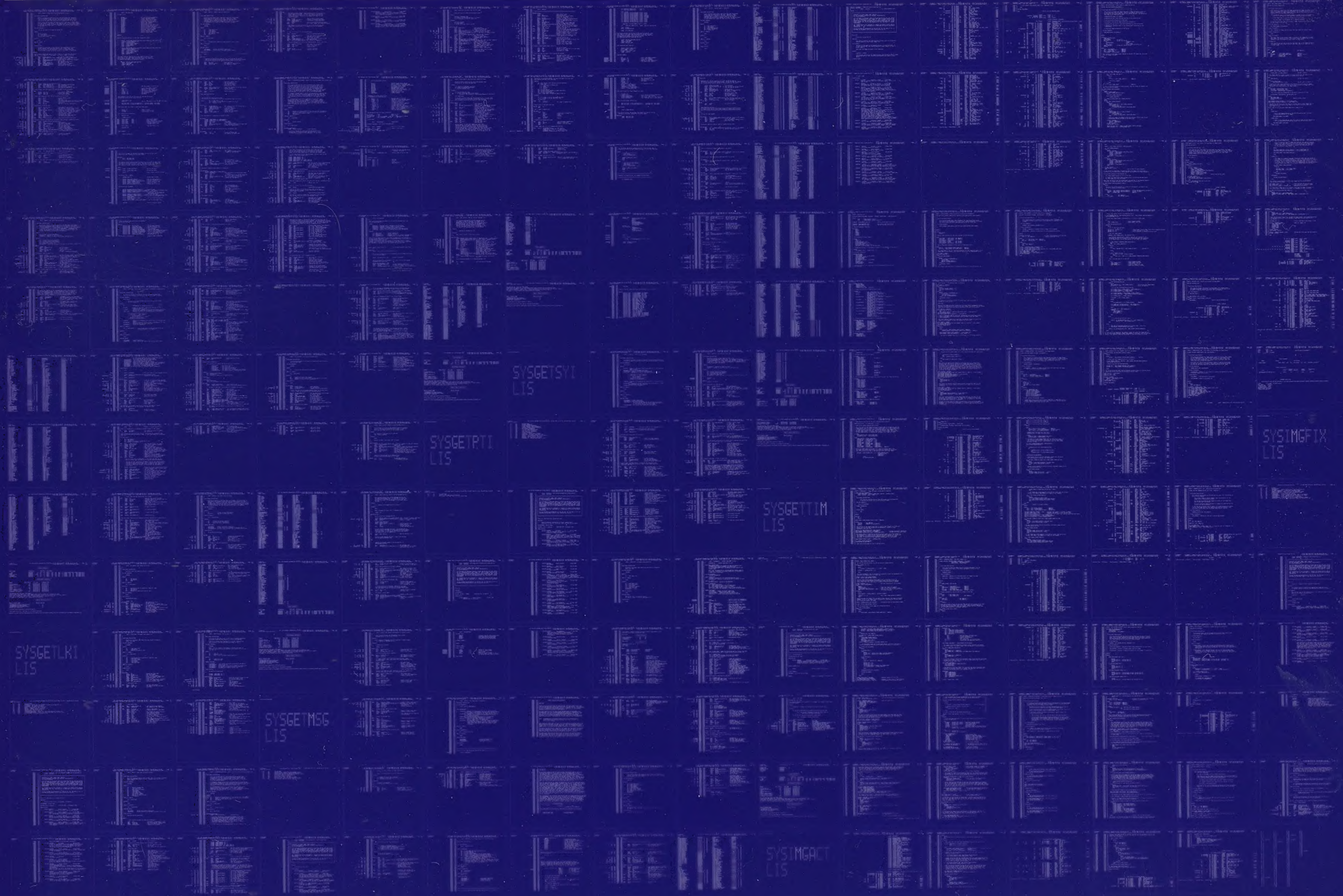
```
SYI$_RMS_PROLOGUE              = 0000108B           SYI$_TTY_SCANDELTA             = 00001073
SYI$_SAVEDUMP                  = 00002013           SYI$_TTY_SILOTIME              = 00001082
SYI$_SBIERRENABLE              = 0000200A           SYI$_TTY_SPEED                 = 00001075
SYI$_SCSBUFFCNT                = 0000105F           SYI$_TTY_TIMEOUT               = 000010F6
SYI$_SCSCONNCNT                = 00001060           SYI$_TTY_TYPAHDSZ              = 0000107B
SYI$_SCSFLOWCUSH               = 00001064           SYI$_UAFALTERNATE              = 0000200E
SYI$_SCSMAXDG                  = 00001062           SYI$_UDABURSTRATE              = 00001070
SYI$_SCSMAXMSG                 = 00001063           SYI$_USER3                     = 0000104A
SYI$_SCSNODE                   = 00001067           SYI$_USER4                     = 0000104B
SYI$_SCSRESPCNT                = 00001061           SYI$_USERD1                    = 00001048
SYI$_SCSSYSTEMID               = 00001065           SYI$_USERD2                    = 00001049
SYI$_SCSSYSTEMIDH              = 00001066           SYI$_VAXCLUSTER                = 000010EE
SYI$_SCS_EXISTS                = 000010DB           SYI$_VERSION                   = 00001000
SYI$_SETTIME                   = 0000200B           SYI$_VIRTUALPAGECNT            = 00001019
SYI$_SID                       = 00001001           SYI$_VMS5                      = 00001044
SYI$_SPTREQ                    = 0000101A           SYI$_VMS6                      = 00001045
SYI$_SRPCOUNT                  = 00001020           SYI$_VMS7                      = 00001046
SYI$_SRPCOUNTV                 = 00001021           SYI$_VMS8                      = 00001047
SYI$_SRPMIN                    = 00001023           SYI$_VMSD1                     = 00001040
SYI$_SRPSIZE                   = 00001022           SYI$_VMSD2                     = 00001041
SYI$_SSINHIBIT                 = 00002011           SYI$_VMSD3                     = 00001042
SYI$_STARTUP_P1                = 000010C2           SYI$_VMSD4                     = 00001043
SYI$_STARTUP_P2                = 000010C3           SYI$_VOTES                     = 000010BD
SYI$_STARTUP_P3                = 000010C4           SYI$_WRITABLESYS               = 0000200F
SYI$_STARTUP_P4                = 000010C5           SYI$_WRITESYSPARAMS            = 00002026
SYI$_STARTUP_P5                = 000010C6           SYI$_WSDEC                     = 00001037
SYI$_STARTUP_P6                = 000010C7           SYI$_WSINC                     = 00001036
SYI$_STARTUP_P7                = 000010C8           SYI$_WSMAX                     = 00001015
SYI$_STARTUP_P8                = 000010C9           SYI$_WS_OPA0                   = 0000202A
SYI$_SWAPFILE_FREE             = 000010F5           SYI$_XFMAXRATE                 = 00001058
SYI$_SWAPFILE_PAGE             = 000010F3           SYI_BIT                        = 00000004
SYI$_SWPALLOCINC               = 0000103C           SYI_S_INCLUSTER                = 00000001
SYI$_SWPFAIL                   = 0000103F           SYI_S_REMOTE_NODE              = 00000001
SYI$_SWPFILCNT                 = 0000100E           SYI_S_RETIRED                  = 00000001
SYI$_SWPOUTPGCNT               = 0000103B           SYI_S_WILD                     = 00000001
SYI$_SWPRATE                   = 0000103A           SYI_V_INCLUSTER                = 00000001
SYI$_SWP_PRIO                  = 0000102E           SYI_V_REMOTE_NODE              = 00000002
SYI$_SYSMWCNT                  = 0000100F           SYI_V_RETIRED                  = 00000003
SYI$_SYSPAGING                 = 0000200D           SYI_V_WILD                     = 00000000
SYI$_SYSPFC                    = 00001004           SYS$DCLAST                       ********  GX   02
SYI$_TAILORED                  = 000010C1           SYS$GB_BRK_LIM                   ********   X   02
SYI$_TBSKIPWSL                 = 00001031           SYS$GB_DEFPRI                    ********   X   02
SYI$_TIMEPROMPTWAIT            = 0000106F           SYS$GB_DEFQUEPRI                 ********   X   02
SYI$_TTY_ALTALARM              = 0000107D           SYS$GB_DFMBC                     ********   X   02
SYI$_TTY_ALTYPAHD              = 0000107C           SYS$GB_DFMBFHSH                  ********   X   02
SYI$_TTY_AUTOCHAR              = 000010F7           SYS$GB_DFMBFIDX                  ********   X   02
SYI$_TTY_BUF                   = 00001078           SYS$GB_DFMBFREL                  ********   X   02
SYI$_TTY_CLASSNAME             = 00001081           SYS$GB_DFMBFSDK                  ********   X   02
SYI$_TTY_DEFCHAR               = 00001079           SYS$GB_DFMBFSMT                  ********   X   02
SYI$_TTY_DEFCHAR2              = 0000107A           SYS$GB_DFMBFSUR                  ********   X   02
SYI$_TTY_DEFPORT               = 00001083           SYS$GB_DFNBC                     ********   X   02
SYI$_TTY_DIALTYPE              = 00001074           SYS$GB_MAXQUEPRI                 ********   X   02
SYI$_TTY_DMASIZE               = 0000107E           SYS$GB_PWD_TMO                   ********   X   02
SYI$_TTY_OWNER                 = 00001080           SYS$GB_RETRY_LIM                 ********   X   02
SYI$_TTY_PARITY                = 00001077           SYS$GB_RETRY_TMO                 ********   X   02
SYI$_TTY_PROT                  = 0000107F           SYS$GB_RMSPROLOG                 ********   X   02
SYI$_TTY_RSPEED                = 00001076           SYS$GL_BRK_TMO                   ********   X   02
```

```
SYS$GL_HID_TIM                   ********   X     02
SYS$GQ_VERSION                   ********   X     02
SYS$GW_BJOBLIM                   ********   X     02
SYS$GW_FILEPROT                  ********   X     02
SYS$GW_GBLBUFQUO                 ********   X     02
SYS$GW_IJOBLIM                   ********   X     02
SYS$GW_NJOBLIM                   ********   X     02
SYS$GW_RJOBLIM                   ********   X     02
SYS$GW_RMSEXTEND                 ********   X     02
TEMPORARY                      = FFFFFFE8
TTY$GB_AUTOCHAR                  ********   X     02
TTY$GB_DEFSPEED                  ********   X     02
TTY$GB_DIALTYP                   ********   X     02
TTY$GB_PARITY                    ********   X     02
TTY$GB_RSPEED                    ********   X     02
TTY$GB_SILOTIME                  ********   X     02
TTY$GL_DEFCHAR                   ********   X     02
TTY$GL_DEFCHAR2                  ********   X     02
TTY$GL_DEFPORT                   ********   X     02
TTY$GL_DELTA                     ********   X     02
TTY$GL_OWNUIC                    ********   X     02
TTY$GL_TIMEOUT                   ********   X     02
TTY$GW_ALTALARM                  ********   X     02
TTY$GW_ALTYPAHD                  ********   X     02
TTY$GW_CLASSNAM                  ********   X     02
TTY$GW_DEFBUF                    ********   X     02
TTY$GW_DMASIZE                   ********   X     02
TTY$GW_PROT                      ********   X     02
TTY$GW_TYPAHDSZ                  ********   X     02
VALUE                          = 00000000
VERIFY_CSB                       00000964 R       02
XTYPE                          = 00000001

                                               +----------------+
                                               ! Psect synopsis !
                                               +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                            00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YF$$SYSGETSYI                    00000BFE  (  3070.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YEXEPAGED                        00000005  (     5.)  03 (  3.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
AEXENONPAGED                     0000000A  (    10.)  04 (  4.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                            +------------------------+
                                            ! Performance indicators !
                                            +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        39    00:00:00.06    00:00:00.59
Command processing                   130    00:00:00.70    00:00:04.42
Pass 1                              1222    00:01:21.37    00:03:18.24
Symbol table sort                      0    00:00:02.71    00:00:06.24
Pass 2                               790    00:00:16.21    00:00:37.57
Symbol table output                   83    00:00:00.52    00:00:01.89
```

```
Psect synopsis output             2      00:00:00.02      00:00:00.02
Cross-reference output            0      00:00:00.00      00:00:00.00
Assembler run totals           2268      00:01:41.60      00:04:08.98

The working set limit was 3000 pages.
439180 bytes (858 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1682 non-local and 67 local symbols.
1428 source lines were read in Pass 1, producing 54 object records in Pass 2.
136 pages of virtual memory were used to define 37 macros.

                                         +----------------------------+
                                         ! Macro library statistics !
                                         +----------------------------+

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1                      5
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          14
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       12
TOTALS (all libraries)                                  31

4130 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSGETSYI/OBJ=OBJ$:SYSGETSYI MSRC$:SYSGETSYI/UPDATE=(ENH$:SYSGETSYI)+EXECML$/LIB+SYS$LIBRARY:SYSBLDMLB/LIB
```

0385 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



SYSGETSYI
LIS

SYSGETPTI
LIS

SYSIMGFIX
LIS

SYSGETTIM
LIS

SYSGETLKI
LIS

SYSGETMSG
LIS

SYSIMGACT
LIS